Department of Computer Science & Engineering

Revised Syllabi (Detailed) for B.Tech in Computer Science and Engineering (2010 Admission onwards)

CS1001 FOUNDATIONS OF COMPUTING

Pre-requisite: NIL

CORE

L	T	P	C
2	0	0	2

Course Outcomes:

CO1: Apply logical inference rules – both propositional and predicate – to decide whether a set of premises result in a given conclusion

CO2: Construct mathematical proofs using mathematical methods of induction, deduction, and contradiction

CO3: Develop recursive solution formulation for computational problems, which can be, modeled recursively.

CO4: Model real life scenarios like transportation networks, social networks etc. using graphs

Total Hours: 28 Hrs

Module 1 : Logic (7 hours)
Propositional logic, implications and inference, equivalence, truth tables. Normal forms. duality, minimization. logic gates and combinational Circuit design, Introduction to first order logic.

Module 2 : Sets and Relational structures (7 hours)
Sets, relations, functions, transitive closures, partial order, lattices, boolean lattices, Boolean algebras.

Module 3 Proof Techniques and Recursion (7 hours)
Methods of proof using Induction, deduction proofs and contradiction. Recursion and recursive definitions. Writing recursive programs.

Module 4 Graphs: (7 hours)
Basic definitions, trees, paths, cycles and elementary properties.

References:

1. E. Mendelson, Shaum's outline on boolean algebra and switching circuits, McGraw Hill, 1970.
2. B. Kolman, R. Busby, R. C. Ross, Discrete Mathematics, Pearson (6/e), 2008.

ZZ1004 COMPUTER PROGRAMMING

Pre-requisite: NIL

CORE

L	T	P	C
2	0	0	2

Course Outcomes:

CO1. Write simple programs in C language.
CO2. Design solutions for simple problems and implement the same using C Language.
CO3. Implement and successfully execute a given algorithm using C language.
CO4. Apply the different fundamental and derived data types for building the solutions.
CO5. Debug syntax errors and logical errors.
CO6. Analyze and find the output of a given C program.

Total Hours: 28 Hrs

Module 1 (7 Hours)
Data Types, Operators and Expressions: Variables and constants - declarations - arithmetic, relational and logical operators – Assignment operator and expressions – conditional expressions – precedence and order of evaluation.
Control Flow: Statements and blocks – if-else, switch, while, for and do-while statements – break and continue statements, goto and labels.

Module 2 (7 Hours)
Functions and Program structure: Basics of functions, Parameter passing – scope rules - recursion.

Module 3 (7 Hours)
Pointers and Arrays: Single and multidimensional arrays - Pointers and arrays – address arithmetic - Passing pointers to functions.

Module 4 (7 Hours)
Structures and Unions: Basics of structures, Structures and functions – Arrays of Structure – Pointers to structures – self referential structures – Type definitions – Unions.
Input and Output: Standard input and output – Formatted output – variable length argument list – file access.

Text Book:

1. B. W. Kernighan and D. M. Ritchie, *The C Programming Language* (2/e), Prentice Hall,1988.

References:

1. B.S. GottFried, *Schaum's Outline of Programming with C*(2/e), McGraw-Hill, 1996.
2. *C. L. Tondo and S. E. Gimpel, The* C Answer Book(2/e), Prentice Hall, 1988.
3. B. W. Kernighan, The Practice of Programming, Addison-Wesley, 1999.

MA2001: MATHEMATICS III

Pre-requisite: MA1001 Mathematics I

CORE

L	T	P	C
3	1	0	3

Total Hours: 56 Hrs

Module 1: Probability distributions (15 Hours)
Random variables, Binomial distribution, Hyper- geometric distribution, Mean and variance of a probability distribution, Chebyshev's theorem, Poisson distribution, Geometric distribution, Normal Distribution, Uniform distribution, Gamma distribution, Beta distribution, Weibull distribution. Joint distribution of two random variables

Module 2: Sampling distributions and Inference concerning means (14 Hours)
Population and samples, The sampling distribution of the mean (σ known and σ unknown), Sampling distribution of the variance, Maximum Likelihood Estimation, Point estimation and interval estimation, point estimation and interval estimation of mean and variance, Tests of hypothesis, Hypothesis concerning one mean, Inference concerning two means.

Module 3: **Inference concerning variances proportions (13Hours)**
Estimation of variances , Hypothesis concerning one variance, Hypothesis concerning two variances , Estimation of proportions , Hypothesis concerning one proportion , Hypothesis concerning several proportions, Analysis of r x c tables, Chi – square test for goodness of fit.

Module 4: Regression Analysis (14 Hours)
Bi-variate Normal distribution- joint, marginal and conditional distributions. Curve fitting, Method of least squares, Estimation of simple regression models and hypothesis concerning regression coefficients, Correlation coefficient- estimation of correlation coefficient, hypothesis concerning correlation coefficient. Estimation of curvilinear regression models,
Analysis of variance:- General principles, Completely randomized designs, Randomized block diagram, Latin square designs, Analysis of covariance.

References:

1. Johnson, R. A., Miller and Freund's Probability and Statistics for Engineers, 6th edition., PHI, 2004.
2. Levin R. I. & Rubin D. S., Statistics for Management, 7th edition, PHI, New Delhi, 2000.
3. S.M. Ross, Introduction to Probability and statistics for Engineers, 3rd edition, Academic Press(Elsevier), Delhi, 2005.

CS2001 LOGIC DESIGN

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Perform the conversion among different number systems.

CO2: Reduce complex logical expressions using various postulates of Boolean algebra.

CO3: Describe graphical methods (like K- Map, Quine-McClusky) for the simplification of complex logical expressions.

CO4: Define and describe basic logic gates.

CO5: Describe design methodology for different combinational logic circuit.

CO6: Describe the structure of various semiconductor programmable logic devices.

CO7: Design concepts of sequential circuits.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Number systems and codes, Boolean algebra: postulates and theorems, constants, variables and functions, switching algebra, Boolean functions and logical operations, Karnaugh map: prime cubes, minimum sum of products and product of sums

Module 2 (10 (T) + 7(P) Hours)
Quine-McClusky algorithm, prime implicant chart, cyclic prime implicant chart, Petrick's method, Combinational Logic: introduction, analysis and design of combinational logic circuits, parallel adders and look-ahead adders, comparators, decoders and encoders, code conversion, multiplexers and demultiplexers, parity generators and checkers

Module 3 (10 (T) + 7(P) Hours)
Programmable Logic Devices, ROMs, PALs, PLAs, PLA folding, design for testability. Introduction to sequential circuits, memory elements, latches

Module 4 (12 (T) + 7(P) Hours)
Flip-flops, analysis of sequential circuits, state tables, state diagrams, design of sequential circuits, excitation tables, Mealy and Moore models, registers, shift registers, counters

References:

1. T. L. Floyd, R. P. Jain, Digital Fundamentals, 8/e, Pearson Education, 2006
2. C. H. Roth, Jr., L. L. Kinney, Fundamentals of Logic Design, 6/e, Cengage Learning, 2009
3. M M Mano, M D Ciletti, Digital Design, 4/e, Pearson Education, 2008
4. N. N. Biswas, Logic Design Theory, Prentice Hall of India, New Delhi, 1993

CS2002 FOUNDATIONS OF PROGRAMMING

Pre-requisite: Nil

CORE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1. Explain concepts and constructs of programming languages.
CO2. Explain and use procedural abstraction, data abstraction, modular design.
CO3. Analyze problems.
CO4. Design correct solutions to problems.
CO5. Write specification of problems and design solutions.

Total Hours: 56 Hrs

Module 1 (14 Hours)
Procedural Abstraction: Expressions - Naming and Environment - Combinators - Evaluation - Procedures - Substitution model - Conditional expression and predicates. Linear Recursion and Iteration - Tree recursion. Abstractions with Higher Order Procedures - Procedures as arguments - Constructing procedures – examples.

Module 2 (14 Hours)
Data Abstraction: Hierarchical Data and Closure property - Symbolic Data - Data Directed Programming - Generic Operators - Combining data of different types

Module 3 (14 Hours)
Modularity, Objects, and State: Local state - assignment, environment model for evaluation - frames, Modeling with mutable data. Concurrency - mechanisms for concurrency. The stream paradigm - modularity.

Module 4 (14 Hours)
Metalinguistic Abstraction: Data as Programs - Separating syntactic analysis from execution. Lazy evaluation - Design of interpreter with lazy evaluation.

References:

1. H Abelson, G J Sussman and J sussman, *Structure and Interpretation of Computer Programs* (2/e), Universities Press, 2005.
2. Companion Site to the Textbook. Available at http://mitpress.mit.edu/sicp/ Accessed on December 1, 2010.

EC2014 SIGNALS AND SYSTEMS

Pre-requisite: Nil

CORE

L	T	P	C
3	0	0	3

Total Hours: 42 Hrs

Module 1 (11 hours)

Elements of signal theory: Different types of signals, basic operations on signals; impulse functions and other singularity functions - Systems : Time-domain representation and analysis of LTI and LSI systems – Convolution - Convolution sum, convolution integral and their evaluation - Causality and stability considerations.

Module 2 (12 hours)

Signal analysis: Signals and vectors – inner product of signals – norm- notion of length of signal and distance between signals– orthogonal signal space – Fourier series representation - Fourier Transform and integral – Fourier Transform theorems – power spectral density and energy spectral density – Hilbert Transform – In-phase and quadrature representation of bandpass signals - Frequency domain analysis of LTI systems: Frequency response Function – signal transmission through a linear system – ideal filters – band width and rise time

Module 3 (8 hours)

Sampling: sampling theorem – sampling with Zero Order Hold and reconstruction – interpolation
Frequency analysis of discrete time signals and systems – Discrete time Fourier series and Discrete time Fourier Transform – Frequency response function – Discrete Fourier Transform.

Module 4 (11 hours)

Laplace transform: Region of convergence – Analysis of continuous time systems – Transfer function – Frequency response from pole – zero plot
Z-transform: Region of convergence – Properties of ROC and Z transform - Analysis of LSI systems - Transfer function- Frequency response from pole – zero plot

References:

1. B. P. Lathi, Linear Systems and Signals, Oxford University Press, 2002.
2. Oppenheim A.V., Willsky A.S. & Nawab S.H., Signals and Systems, Second edition , Tata McGraw Hill
3. Haykin S. & Veen B.V., Signals & Systems,1999, John Wiley
4. Taylor F.H., Principles of Signals & Systems,1994, McGraw Hill

CS2091 LOGIC DESIGN LABORATORY

Pre-requisite: Nil

CORE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1: Describe basic logic gates.

CO2: Design and implement combinational circuits.

CO3: Design and implement sequential circuits.

Total Hours: 56 Hrs

Theory (14 Hours)

Logic gates, adder and subtractor circuits, parity generators, code converters, comparators, multiplexers, demultiplexers, flip-flops, shift registers, counters

Practical (42 Hours)

Design and implementation of logic gates, adder and subtractor circuits, parity generators, code converters, comparators, multiplexers, demultiplexers, flip-flops, shift registers, counters

References:

1. C H Roth and Jr., L L Kinney, Fundamentals of Logic Design, 6/e, Cengage Learning, 2009
2. M M Mano and M D Ciletti, Digital Design, 4/e, Pearson Education, 2008
3. N N Biswas, Logic Design Theory, Prentice Hall of India, New Delhi, 1993
4. T L Floyd and R P Jain, Digital Fundamentals, 8/e, Pearson Education, 2006

CS2092 PROGRAMMING LABORATORY

Pre-requisite: Nil

CORE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Explain and apply procedural abstraction, data abstraction, and modular design.
CO2. Design correct solutions to problems.
CO3. Write program in language Scheme.
CO4. Write reliable programs.

Total Hours: 56 Hrs

Theory (14 Hours)

Introduction to the language of choice (recommended: Scheme). Overview of concepts and constructs.
Study of synchronization aspects. Interpreter specification.

Practical (42 Hours)

Programming Assignments

1. Simple programs in the language of choice - (recommended Scheme) - evaluating expressions.
2. Programming example with procedures - Operations.
3. Introduction to syntax, semantics and symbolic manipulation in the language.
4. Combining data and procedural abstractions – Objects.
5. Synchronization and Concurrency examples.
6. Design of a simple language interpreter.

<u>References:</u>

1. H Abelson, G J Sussman and J sussman, *Structure and Interpretation of Computer Programs* (2/e), Universities Press, 2005.

2. Sample Programming Assignments from Reference 1. Available at http://mitpress.mit.edu/sicp/psets/index.html Accessed on December 1, 2010.

MA2002 MATHEMATICS IV

Pre-requisite: MA 1001 Mathematics I, MA 1002 Mathematics II

CORE

L	T	P	C
3	1	0	3

Total Hours: 56 Hrs

Module 1: Series Solutions and Special Functions (15 Hours)
Power series solutions of differential equations, Theory of power series method, Legendre Equation, Legendre Polynomials, Frobenius Method, Bessel's Equation, Bessel functions, Bessel functions of the second kind, Sturm- Liouville's Problems, Orthogonal eigenfunction expansions.

Module 2: Partial differential Equations (16 Hours)
Basic Concepts, Cauchy's problem for first order equations, Linear Equations of the first order, Nonlinear Partial Differential Equations of the first order, Charpit's Method, Special Types of first order equations, Classification of second order partial differential equations, Modeling: Vibrating String, Wave equation, Separation of variables, Use of Fourier Series, D'Alembert's Solution of the wave equation, Heat equation: Solution by Fourier series, Heat equation: solution by Fourier Integrals and transforms, Laplace equation, Solution of a Partial Differential Equations by Laplace transforms.

Module 3: Complex Numbers and Functions (13 Hours)
Complex functions, Derivative , Analytic function, Cauchy- Reimann equations, Laplace's equation, Geometry of Analytic functions: Conformal mapping, Linear fractional Transformations, Schwarz - Christoffel transformation, Transformation by other functions.

Module 4: Complex Integration (12 Hours)
Line integral in the Complex plane, Cauchy's Integral Theorem, Cauchy's Integral formula, Derivatives of analytic functions.Power series, Functions given by power series, Taylor series and Maclaurin's series. Laurent's series, Singularities and Zeros, Residue integration method, Evaluation of real Integrals.

References:

1. Kreyszig E, Advanced Engineering Mathematics, 8^{th} Edition, John Wiley & Sons, New York, 1999 .
2. I.N. Sneddon, Elements of Partial Differential Equations, Dover Publications, 2006.
3. Wylie C. R. & Barret L. C., Advanced Engineering Mathematics, 6^{th} Edition, Mc Graw Hill, New York,1995.
4. Donald W. Trim, Applied Partial Differential Equations, PWS – KENT publishing company, 1994.

CS2004 COMPUTER ORGANIZATION

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Describe the hardware components of a computer and its architecture

CO2. Specify the steps for evaluating and comparing the performance of computers

CO3. Specify the instruction set of MIPS architecture to build a basic processor

CO4. Design and construct a basic processor using single cycle, multicycle, pipelined techniques

CO5. Analyze the performance problems and give solutions in pipelined processor implementations.

CO6. Analyze and specify new memory interactions to improve the performance of a computing system.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Computer abstraction and technology: basic principles, hardware components, Measuring performance: evaluating, comparing and summarizing performance.

Instructions: operations and operands of the computer hardware, representing instructions, making decision, supporting procedures, character manipulation, styles of addressing, starting a program.

Module 2 (10 (T) + 7(P) Hours)

Computer arithmetic: signed and unsigned numbers, addition and subtraction, logical operations, constructing an ALU, multiplication and division, floating point representation and arithmetic, Parallelism and computer arithmetic.

Module 3 (10 (T) + 7(P) Hours)

The processor: building a data path, simple and multicycle implementations, microprogramming, exceptions, Pipelining, pipeline data path and Control , hazards in pipelined processors

Module 4 (12 (T) + 7(P) Hours)

Memory hierarchy: caches, cache performance, virtual memory, common framework for memory hierarchies

Input/output: I/O performance measures, types and characteristics of I/O devices, buses, interfaces in I/O devices, design of an I/O system, parallelism and I/O.

References:

1. D. A. Pattersen and J. L. Hennesy, Computer Organisation and Design: The Hardware/ Software Interface, 4/e, Morgan Kaufman, 2009.

2. V. P. Heuring and H. F. Jordan, Computer System Design and Architecture, Prentice Hall, 2003.

CS2005 DATA STURCTURES AND ALGORITHMS

Pre-requisite: Nil

CORE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1. Define and describe simple data structures like arrays, linked lists, trees and graphs

CO2. Design and specify algorithms for searching and sorting, and those associated with the above data structures

CO3. Analyze simple algorithms, like sorting and searching using mathematical tools, like formulation and solving of recurrences, asymptotic analysis and probabilistic analysis

CO4. Analyze application problems and abstract them to formulate solutions involving data structures and algorithms

CO5. Analyze the correctness of algorithms

Total Hours: 56 Hrs

Module 1 (14 Hours)

Time and space complexity analysis of algorithms - Asymptotic analysis - Big Oh - Omega - theta notations - Searching and Sorting - Binary search - Quick sort - Heap sort - priority queue using heap - complexity analysis of search and sorting algorithms - average case analysis of quick sort.

Module 2 (14 Hours)

Linked lists - Stack and Queue - Binary tree - in-order, pre-order and post-order traversals - complexity analysis - representation and evaluation of arithmetic expressions using binary tree - Binary Search trees - insertion, deletion and search - average case complexity analysis.

Module 3 (14 Hours)

File structure - Merge sort - B Tree - complexity analysis - Data structures for disjoint sets - union by rank and path compression - complexity analysis - Hash tables.

Module 4 (14 Hours)

Graph representation- DFS, BFS, minimum spanning tree problem - Kruskal's algorithm - implementation using disjoint set data structure - complexity analysis – Prim's algorithm - Shortest path problem - Dijkstra's algorithms - implementation of Prim's and Dijkstra's algorithms using priority queue data structure - complexity analysis. Floyd-Warshall algorithm.

References:

1. T. H. Cormen, C. E. Lieserson, R. L. Rivest, C. Stein, Introduction to Algorithms (3/e), MIT Press, 2003
2. S. Dasgupta, C. H. Papadimitriou, U. Vazirani, Algorithms, McGraw Hill, 2006
3. A. V. Aho, J. D. Ullman and J. E. Hopcroft, Data Structures and Algorithms, Addison Wesley, 1983.

CS2006 DISCRETE STRUCTURES

Pre-requisite: Nil

CORE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1: Solve a given recurrence relation and express the solutions in asymptotic notation.

CO2: Develop appropriate probabilistic model for a given problem of algorithmic nature and computation of its statistical parameters.

CO3: Express and solve number theoretic problems using algebraic properties of groups, rings and fields.

CO4: Distinguish between countable and uncountable sets and formally establish the cardinality of a given set.

CO5: Develop a discrete model for a given computational problem and solve.

Total Hours:56 Hrs

Module 1 (14 Hours)
Combinatorics: Asymptotic analysis of recurrence - solution to linear recurrence relations - Master's theorem, Recurrence relations with full history.

Module 2 (14 Hours)
Probability: Discrete probability spaces, random variables - Bernoulli, binomial and geometric random variables - conditional probability - Bayes theorem - linearity of expectations - Markov and Chebyshev inequalities - weak law of large numbers

Module 3 (14 Hours)
Algebra: Groups, Lagrange's theorem, Homomorphism theorem, Rings and Fields, Structure of the ring Zn and the unit group Zn*.

Module 4 (14 Hours)
Logic and Set Theory: Resolution in propositional logic - introduction to first order logic - set theory - countable and uncountable sets - diagonalization.

References:

1. R. P. Grimaldi, Discrete and Combinatorial Mathematics: An Applied Introduction, Addison Wesley, 1998.
2. L. Lovasz, J. Pelikan and K. Vesziergombi, Discrete Mathematics, Springer, 2003.
3. I. M. Copi, Symbolic logic, Prentice Hall, 1979

CS2093 HARDWARE LABORATORY

Pre-requisite: Nil

CORE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Describe the x86 architecture, its instruction set and basics of NASM Assembler

CO2. Use software interrupts for implementing input output operations in NASM programs

CO3. Design programs using integer operations.

CO4. Develop programs in NASM using string related instructions

CO5. Design and implement programs using 80x86 floating point instructions

Total Hours: 56 Hrs

Theory (14 Hours) + Practical (42 Hours)

80X86 Assembly language programming:

Integer operations, recursive subroutines, two dimensional arrays (3(T) +12(P) Hours)

String manipulation, floating point operations (2(T) + 6(P) Hours)

DOS and BIOS interrupts. (2(T) + 6(P) Hours)

Embedded system experiments (RTLinux). (3(T) + 9(P) Hours)

Cache simulator – Performance evaluation of various cache organizations optimizations (2(T) +6(P) Hours)

Familiarization of PC hardware and trouble shooting (2(T) +3(P) Hours)

References:

1. Peter Abel IBM PC Assembly Language and Programming (5/e), Prentice Hall, 2001.

2. Barry B Brey, Intel Microprocessors: Architecture and Programming, Prentice Hall, 2008.

CS2094 DATA STRUCTURES LABORATORY

Pre-requisite: Nil

CORE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Implement fundamental algorithms like sorting and searching and simple data structures like arrays, linked lists, trees and graphs

CO2. Analyze the computing problems given and assess the suitability of different data structures and algorithms to solve the problems

CO3. Design and implement algorithmic solutions within suitable time constraints to face real life practical situations in the computing industry by following the ethics of computing

Total Hours: 56 Hrs

Theory (14 Hours)

Review of dynamic memory allocation - use of pointers - review of recursion. File organization.

Practical (42 Hours)

1. Searching: Binary search implementation
2. Sorting: Heap sort, Quick sort and Merge sort implementation
3. Stack and Queue implementation using linked list
4. Arithmetic expression to postfix
5. Postfix to expression tree, tree traversal and evaluation
6. Binary search tree - insert, delete and search
7. Linear time DFS and BFS implementation with adjacency list representation
8. Kruskal's algorithm implementation in O((n+e)log n) complexity.
9. Prim's algorithm implementation in O((n+e) log n) complexity.

10 Dijskstra's algorithm implementation in O((n+e) log n) complexity.

References:

1. T. H. Cormen, C. E. Lieserson and R. L. Rivest, Introduction to Algorithms, PHI, 1998
2. S. Sahni, Data structures, Algorithms, and Applications in C++, McGraw Hill, 1998

CS3001 THEORY OF COMPUTATION

Pre-requisite: Nil

CORE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1. Design finite state machines, regular expressions and grammars for given languages

CO2. Prove the equivalence of languages described by automata and grammars

CO3. Analyze various models of computation and fundamental limits of computation

CO4. Classify problems and place them in the right level of complexity

CO5: Apply NP-completeness concepts to create proofs regarding the computational complexity of novel problems

Total Hours: 56 Hrs

Module 1 (14 Hours)

Basic concepts of Languages, Automata and Grammar. Regular Languages - Regular expression - finite automata equivalence, Myhill Nerode theorem and DFA State Minimization, Pumping Lemma and proof for existence of non-regular languages.

Module 2 (14 Hours)

Context Free languages, CFL-PDA equivalence, Pumping Lemma and proof for existence of non-Context Free languages, CYK Algorithm, Deterministic CFLs.

Module 3 (14 Hours)

Turing Machines: recursive and recursively enumerable languages, Universality of Turing Machine, Church Thesis. Chomsky Hierarchy, Undecidability, Reducibility Undecidability: Recursive and Recursively enumerable sets, Rice Theorems., Recursion Theorem, Turing Reducibility, Hierarchy theorems,

Module 4 (14 Hours)

Complexity: P, NP, NP Completeness, PSPACE and Log space. Logic: Propositional logic, compactness, Decidability, Resolution

References:

1. M. Sipser, *Introduction to the Theory of Computation*, Thomson, 2001.
2. C. H. Papadimitriou., Computational *Complexity*, Addison Wesley, 1994.
3. Jerome Keisler H. Joel Robbin, Mathematical Logic and Computability, McGraw-Hill International Editions, 2000.
4. C. H. Papadimitriou, H. Lewis, Elements of Theory of Computation, Prentice Hall, 1981.
5. J. E. Hopcroft R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison Wesley, 3/e, 2006.
6. J. C. Martin, Introduction to Languages and the Theory of Computation, Mc Graw Hill, 2002.
7. M. R. Garey and D. S. Johnson. Computers & Intractability, W. H. Freeman & Co., San Francisco, 1979.
8. S. M. Srivastava, A Course on Mathematical Logic, Springer, 2008.

CS3002 DATABASE MANAGEMENT SYSTEMS

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Model, design and normalize databases for real life applications.

CO2: Code and deploy databases for applications using RDBMS like ORACLE

CO3: Query Database applications using Query Languages like SQL, QBE

CO4: Undertake and successfully complete Database Development projects within the allotted time.

CO5: Deploy efficient IT solutions using free and open software and help the society

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Database System concepts and architecture, Data modeling using Entity Relationship (ER) model and Enhanced ER model, Specialization, Generalization, Data Storage and indexing, Single level and multi level indexing, Dynamic Multi level indexing using B Trees and B+ Trees.

Module 2 (10 (T) + 7(P) Hours)

The Relational Model, Relational database design using ER to relational mapping, Relational algebra and relational calculus, Tuple Relational Calculus, Domain Relational Calculus, SQL.

Module 3 (10 (T) + 7(P) Hours)

Database design theory and methodology, Functional dependencies and normalization of relations, Normal Forms, Properties of relational decomposition, Algorithms for relational database schema design.

Module 4 (12 (T) + 7(P) Hours)

Transaction processing concepts, Schedules and serializability, Concurrency control, Two Phase Locking Techniques, Optimistic Concurrency Control, Database recovery concepts and techniques, Introduction to database security.

References:

1. Ramez Elmasri and Shamkant B. Navathe, Fundamentals of Database Systems (5/e), Pearson Education, 2008.
2. Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems (3/e), McGraw Hill, 2003.
3. Peter Rob and Carlos Coronel, Database Systesm- Design, Implementation and Management (7/e), Cengage Learning, 2007.

CS3003 OPERATING SYSTEMS

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Describe the working of a computing system from hardware to application program.

CO2. Describe the design of a basic operating system

CO3. Define processes, threads, interprocess and thread communication mechanisms and synchronization techniques.

CO4. Describe the kernel functions - process management, memory management, device management and file management

CO5. Construct different OS modules.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Review of operating system strategies - resources - processes - threads - objects - operating system organization - design factors - functions and implementation considerations - devices - characteristics - controllers - drivers - device management - approaches - buffering - device drivers - typical scenarios such as serial communications - storage devices etc

Module 2 (10 (T) + 7(P) Hours)

Process management - system view - process address space - process and resource abstraction - process hierarchy - scheduling mechanisms - various strategies - synchronization - interacting & coordinating processes - semaphores - deadlock - prevention - avoidance - detection and recovery

Module 3 (10 (T) + 7(P) Hours)

Memory management - issues - memory allocation - dynamic relocation - various management strategies - virtual memory - paging - issues and algorithms - segmentation - typical implementations of paging & segmentation systems

Module 4 (12 (T) + 7(P) Hours)

File management - files - implementations - storage abstractions - memory mapped files - directories and their implementation - protection and security - policy and mechanism - authentication - authorization - case study of Unix kernel and Microsoft Windows NT (concepts only)

Virtual machines – virtual machine monitors – issues in processor, memory and I/O virtualization, hardware support for virtualization.

References:

1. Silberschatz, Galvin and Gagne, Operating System Principles, 7/e, 2006, John Wiley
2. William Stallings, Operating Systems, 5/e, Pearson Education
3. Crowley C., Operating Systems- A Design Oriented Approach, Tata McGraw Hill, New Delhi
4. Tanenbaum A. S., Modern Operating Systems, 3/e Prentice Hall, Pearson Education
5. Gary J. Nutt, Operating Systems - A Modern Perspective,3/e Addison Wesley

CS3004 SOFTWARE ENGINEERING

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Apply the basic concepts, principles and theories in software engineering to build software systems from the scratch, considering both technical and managerial aspects.

CO2. Design and implement different phases in the life cycle of software development and identify appropriate process models.

CO3. Analyze real problems/requirements and design systems by developing specifications and abstractions to make development of complex systems easy.

CO4. Create well-structured software documentation /deliverables according to industry standards at each mile stones by making use of formal techniques and specification.

CO5. Use standard metrics/techniques for software systems for cost/effort estimation, project scheduling, risk analysis and management, configuration management and quality assurance

Total Hours: 70 Hrs

Module 1 (8 (T) + 7(P) Hours)

Introduction to Software Engineering – Reasons for software project failure – Similarities and differences between software and other engineering products.

Software Development Life Cycle (SDLC) – Overview of Phases.

Detailed Study of Requirements Phase: Importance of Clear Specification – Formal specification methods including algebraic specification in detail.

Module 2 (15 (T) + 7(P) Hours)

Problem partitioning (subdivision) - Power of Abstraction

Concept of functional decomposition – process modeling - DFDs

Concept of data modeling – ER diagrams

Class and component level designs – Design - UML and Design Patterns (only introduction)

Module 3 (8 (T) + 7(P) Hours)

Coding and Testing :

Structured programming – internal documentation and need for standards – Methods of version control - Maintainability.

Introduction to secure programming.

Types of testing – Specification of test cases – Code review process

Module 4 (11 (T) + 7(P) Hours)

Software Project Management: Introduction to metrics. Software Process Models. Costing, Scheduling and Tracking techniques. Software configuration management - versioning. Reusable components. Mathematical methods of risk assessment and management. Methods of software licensing and introduction to free software.

References:

1. Roger S Pressman, Software Engineering: A Practitioner's Approach (6/e.), McGraw Hill, 2008.
2. T C Lethbridge and R Laganiere, Object Oriented Software Engineering (1/e), Tata McGraw Hill, 2004.
3. Pankaj Jalote, Software Engineering: A Precise Approach (1/e), Wiley India, 2010.
4. A Shalloway and J Trott, Design Patterns Explained: A new perspective on object oriented design (2/e), Pearson,

2004.

CS3005 COMPILER DESIGN

Pre-requisite: CS2005 Data Structures and Algorithms

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Define and explain syntax and semantics of programs.

CO2. Define and explain syntax and semantics analysis of programs.

CO3. Design a compiler.

CO4. Analyze complex problems.

CO5. Design complex systems.

CO6. Synthesize a compiler.

Total Hours: 70 Hrs

Module 1 (6 (T) + 7(P) Hours)

Introduction to Programming language translation. Lexical analysis: Specification and recognition of tokens.

Module 2 (12 (T) + 7(P) Hours)

Syntax analysis: Top-down parsing-Recursive descent and Predictive Parsers. Bottom-up Parsing- LR (0), SLR, and LR (1) Parsers.

Module 3 (16 (T) + 7(P) Hours)

Semantic analysis: Type expression, type systems, symbol tables and type checking.

Intermediate code generation: Intermediate languages. Intermediate representation-Three address code and quadruples. Syntax-directed translation of declarations, assignments statements, conditional constructs and looping constructs.

Module 4 (8 (T) + 7(P) Hours)

Runtime Environments: Storage organization, activation records. Introduction to machine code generation and code optimizations.

References:

1. Aho A.V., Lam M. S., Sethi R., and Ullman J. D., Compilers: Principles, Techniques and Tools, Pearson Education, 2007.

2. Appel A.W, and Palsberg J. , Modern Compiler Implementation in Java, Cambridge University Press, 2002.

CS3006 COMPUTER NETWORKS

Pre-requisite: Nil

CORE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Define and describe functionality and services offered at various layers of TCP/IP protocol stack like Data Link Layer, Network Layer, Transport Layer and Application layer.

CO2. Analyze the specification and implement important protocols in each layer of the TCP/IP stack.

CO3. Analyze and solve problems with respect to various layers in the TCP/IP stack using mathematical tools.

CO4. Apply tools like Wireshark and ns2 to design and/or analyze performance of a network topology.

CO5. Design and implement simple application based on socket programming

CO6: Correlate the real life scenarios with the concepts learned in computer networking

.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Computer Networks and Internet, the network edge, the network core, network access, delay and loss, protocol layers and services, Application layer protocols, Web 2.0, Socket Programming,

Module 2 (10 (T) + 7(P) Hours)

Transport layer services, UDP, TCP, New transport layer Protocols, congestion control, new versions of TCP, Network layer services, routing, IP, routing in Internet, router, IPV6, multicast routing.

Module 3 (10 (T) + 7(P) Hours)

Link layer services, error detection and correction, multiple access protocols, ARP, Ethernet, hubs, bridges, switches, wireless links, mobility, PPP, ATM, MPLS, VLAN.

Module 4 (12 (T) + 7(P) Hours)

Multimedia networking, streaming stored audio and video, real-time protocols, security, Cryptography, authentication, integrity, key distribution, network management.

References:

1. J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach Featuring Internet, 3/e, Pearson Education, 2005.
2. Peterson L.L. & Davie B.S., Computer Networks, A systems approach, 3/E, Harcourt Asia, 2003.
3. Andrew S. Tanenbaum, Computer Networks, 3/E, PHI, 1996.
4. Adrian Farrel, The Internet and its Protocols a Comparative Approach, Elsevier, 2005
5. IEEE/ACM Transactions on Networking

CS4001 ENVIRONMENTAL STUDIES

Pre-requisite: Nil

CORE

L	T	P	C
3	0	0	3

Course Outcomes:

CO1. Define and describe a system component or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety and sustainability.

CO2. State the fundamental environmental science and engineering principles necessary to solve equitable use of resources and sustainable lifestyles.

CO3. Analyze environmental pollution and methods of prevention like air, water, soil, marine, thermal, nuclear and noise pollution and waste management.

CO4. Analyze human population and human rights

Total Hours: 42 Hrs

Module 1 (10 Hours)

Definition, scope and importance - renewable and non-renewable resources - Natural resources - forest, water, mineral, food and energy and land resources - study of problems - Role of individual in conservation - equitable use of resources and sustainable lifestyles.

Module 2 (10 Hours)

Eco systems - structure and function - producer, consumer and decomposer - energy flow - ecological succession- food chains- forest, grassland, desert and aquatic ecosystems - Biodiversity and conservation.

Module 3 (10 Hours)

Environmental pollution - air, water, soil, marine, thermal, nuclear and noise pollution- methods of prevention - waste management - disaster management - environmental ethics - sustainable development models - water conservation - climate change and global warming - ozone layer depletion - nuclear holocaust - case studies - consumerism and waste products.

Module 4 (12 Hours)

Human population and environment - family welfare - human health and environment - human rights.

References:

1. E. Bharucha, Environmental Studies, Universities Press, 2005.
2. UGC Syllabus on environmental studies available at http://www.ugc.ac.in/inside/syllabus.html accessed on 01-12-2010

MS4003 ECONOMICS

Pre-requisite: Nil

CORE

L	T	P	C
3	0	0	3

Total Hours: 42 Hrs

Module 1 (11 hours)

General Foundations of Economics; Nature of the firm; Forms of organizations-Objectives of firms-Demand analysis and estimation-Individual, Market and Firm demand, Determinants of demand, Elasticity measures and business decision making, Theory of the firm-Production functions in the short and long run

Module 2 (9 hours)

Cost concepts- Short run and long run costs- economies and diseconomies of scale, real and pecuniary economies; Product Markets; Market Structure- Competitive market; Imperfect competition (Monopoly, Monopolistic & Oligopoly) and barriers to entry and exit -Pricing in different

Module 3 (11 hours)

Macro Economic Aggregates-Gross Domestic Product; Economic Indicators; Models of measuring national income; Inflation ; Fiscal and Monetary Policies ; Monetary system; Money Market, Capital market; Indian stock market; Development Banks; Changing role of Reserve Bank of India

Module 4 (11 hours)

International trade - Foreign exchange market- Balance of Payments (BOP) and Trade-Effects of disequilibrium in BOP in business- Trade regulation- Tariff versus quotas- International Trade and development and role of international institutions (World Bank, IMF and WTO) in economic development.

References

1. Bo Soderston,International Economics,
2. Gupta, S.B Monetary Economics,. (1994). S. Chand & Co., New Delhi.
3. Gregory.N.Mankiw,Principles of Micro Economics, Cengage Publications,2007
4. Gregory.N.Mankiw ,Principles of Macro Economics, Cengage Publications,2007
5. *Indian Economy – Its Development Experience*, Misra, S.K. and V.K. Puri (2001)Himalaya Publishing House, Mumbai,2009.
6. *Microeconomics*, R.S. Pindyck, D.L Rubinfield and P.L. Mehta ,Pearson Education, 2005. *Advanced Economic Theory*, Micro Economics H.L. Ahuja,Chand Publications,2004.
7. Economics, Samuelson, P.A.;& W.D. Nordhaus , Tata McGraw Hill,18 Ed.,2005.
8. Public Finance , B.P.Tyagi,Jai PrakashNath & Co.,1997.

ME4104 PRINCIPLES OF MANAGEMENT

Prerequisite: Nil

CORE

L	T	P	C
3	0	0	3

Total Hours: 42 hours

Module 1 (9 Hours)
Introduction to management theory, Characteristics of management, Management as an art – profession, Systems approach to management, Task and responsibilities of a professional manager, Levels of managers and skill required. Management process – planning – mission – objectives – goals – strategy – policies – programmes – procedures.

Module 2 (9 Hours)
Organizing – principles of organizing – organization structures, Directing – delegation – span of control – leadership – motivation – communication, Controlling.

Module 3 (12 Hours)
Decision making process – decision making under certainty – risk – uncertainty – models of decision making, Project management – critical path method – programme evaluation and review technique – crashing.

Module 4 (12 Hours)
Introduction to functional areas of management, Operations management, Human resources management, Marketing management, Financial management.

References

1. Koontz, H., and Weihrich, H., *Essentials of Management: An International Perspective*, 8th ed., McGraw Hill, 2009.
2. Hicks, *Management: Concepts and Applications*, Cengage Learning, 2007.
3. Mahadevan, B., *Operations Management, Theory and Practice*, Pearson Education Asia, 2009.
4. Kotler, P., Keller, K.L, Koshy, A., and Jha, M., *Marketing Management*, 13th ed., 2009.
5. Khan, M.Y., and Jain, P.K., *Financial Management*, Tata-Mcgraw Hill, 2008.

CS4021 NUMBER THEORY AND CRYPTOGRAPHY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Solve number theoretic problems and understand their role in cryptosystems.

CO2. Apply cryptographic techniques for encryption, hashing, signature and authentication.

CO3. Construct mathematical arguments about the security of the cryptosystem.

CO4. Deploy the cryptographic techniques to detect and prevent basic security threats.

Total Hours: 70 Hrs

Module 1 (8 (T) + 7(P) Hours)

Divisibility theory in integers. Extended Euclid's algorithm. Modular Arithmetic – exponentiation and inversion. Fermat's Little Theorem, Euler's Theorem. Solution to congruences, Chinese Remainder Theorem.

Module 2 (12 (T) + 7(P) Hours)

Review of abstract algebra – Study of Ring Zn, multiplicative group Zn* and finite field Zp – Gauss Theorem (cyclicity of Zp*) - Quadratic Reciprocity.

Primality Testing – Fermat test, Carmichael numbers, Solovay Strassen Test, Miller Rabin Test - analysis.

Module 3 (13 (T) + 7(P) Hours)

Notions of security. Introduction to one secret key cryptosystem (DES) and one cryptographic hash scheme (SHA).

Public Key Cryptosystems – Diffie Hellman Key Agreement Protocol, Knapsack crypto systems, RSA. Elgamal's encryption and signature scheme.

Module 4 (9 (T) + 7(P) Hours)

Authentication Protocols: One way and Mutual Authentication, Challenge Response protocols, Lamport's scheme, Needham Schroeder protocol. Interactive proof systems, Zero Knowledge Proof systems – soundness and completeness – Fiat-Shamir identification scheme.

References:

1. H. Delfs and H. Knebl, Introduction to Cryptography: Principles and Applications, Springer-Verlag, 2002.
2. Serge Vaudney, A Classical Introduction to Cryptography: Applications for Communications Security, Springer, 2009.
3. Bernard Menezes, Network Security and Cryptography. Cengage Learning, 2010.
4. B A Forouzan and D Mukhopadyay, Cryptography and Network Security(2/e). Tata McGraw Hill, 2010

CS4022 PRINCIPLES OF PROGRAMMING LANGUAGES

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Design Programming Language features like semantics, and type systems
CO2. Specify the formal semantics of programming languages
CO3. Write precise mathematical statements about properties of programming languages
CO4. Design safe type systems
CO5. Use automatic program development tools like Lexical analyzer / Parser generator
CO6. Design and implement programming language interpreter
CO7. Prove the properties of programming languages like type safety

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Programming Languages: Concepts and Constructs. Untyped Arithmetic Expressions – Introduction, Semantics, Evaluation.

Module 2 (10 (T) + 7(P) Hours)
Untyped Lambda Calculus – Basics, Semantics. Programming in Lambda Calculus.

Module 3 (10 (T) + 7(P) Hours)
Typed Arithmetic Expressions – Types and Typing relations, Type Safety.
Simply Typed Lambda Calculus – Function types, Typing relations, Properties of typing.

Module 4 (12 (T) + 7(P) Hours)
Extensions to Simply Typed Lambda Calculus – Unit type, Let bindings, Pairs, Records, Sums, Variants, References, Exceptions.

References:

1. Benjamin C. Pierce, Types and Programming Languages , MIT Press, 2002
2. David A. Schmidt, Programming Language Semantics. In Allen B. Tucker, Ed. Handbook of Computer Science and Engineering, CRC Press, 1996.
3. Luca Cardelli, Type Systems. In Allen B. Tucker, Ed. Handbook of Computer Science and Engineering, CRC Press, 1996.
4. Michael L. Scott, Programming Language Pragmatics, Elsevier (2/e), 2004

CS4023 COMPUTATIONAL INETELLIGENCE

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

Total Hours: 70 Hrs

Module 1 (10(T) + 7(P) Hours)

Artificial Intelligence: History and Applications, Production Systems, Structures and Strategies for state space search- Data driven and goal driven search, Depth First and Breadth First Search, DFS with Iterative Deepening, Heuristic Search- Best First Search, A* Algorithm, AO* Algorithm, Local Search Algorithms and Optimization Problems, Constraint satisfaction, Using heuristics in games- Minimax Search, Alpha Beta Procedure. Implementation of Search Algorithms in LISP.

Module 2 (10(T) + 7(P) Hours)

Knowledge representation - Propositional calculus, Predicate Calculus, Forward and Backward chaining, Theorem proving by Resolution, Answer Extraction, AI Representational Schemes- Semantic Nets, Conceptual Dependency, Scripts, Frames, Introduction to Agent based problem solving. Implementation of Unification, Resolution and Answer Extraction using Resolution.

Module 3 (10(T) + 7(P) Hours)

Machine Learning- Symbol based and Connectionist, Social and Emergent models of learning, Planning-Planning and acting in the real World, The Genetic Algorithm- Genetic Programming, Overview of Expert System Technology- Rule based Expert Systems, Introduction to Natural Language Processing. Implementation of Machine Learning algorithms.

Module 4 (12(T) + 7(P) Hours)

Languages and Programming Techniques for AI- Introduction to PROLOG and LISP, Search strategies and Logic Programming in LISP, Production System examples in PROLOG.

References:

1. George F Luger, Artificial Intelligence- Structures and Strategies for Complex Problem Solving, 4/e, 2002, Pearson Education.
2. E. Rich and K.Knight, Artificial Intelligence, 2/e, Tata McGraw Hill
3. S Russel and P Norvig, Artificial Intelligence- A Modern Approach, 2/e, Pearson Education, 2002
4. Nils J Nilsson, Artificial Intelligence a new Synthesis, Elsevier,1998
5. Winston. P. H, LISP, Addison Wesley
6. Ivan Bratko, Prolog Programming for Artificial Intelligence, 3/e, Addison Wesley, 2000
7. Dr.Russell Eberhart and Dr.Yuhui shi, Computational Intelligence - Concepts to Implementation, Elsevier, 2007
8. Fakhreddine O Karray, Clarence De Silva, Soft Computing and Intelligent Systems Design- Theory tools and Applications, Pearson Education, 2009.

CS4024 INFORMATION THEORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
4	0	0	4

.

Total Hours: 56 Hrs

Module 1 (14 Hours)
Foundations: Review of probability theory, entropy and information, random sources, i.i.d and Markov sources, discrete finite state stationary Markov sources, Entropy rate of stationary sources, Computation of stationary distributions.

Module 2 (14 Hours)
Source Coding: Prefix and uniquely decodable codes - Kraft's and Macmillan's inequalities - Shannon's source coding theorem - Shannon Fano code, Huffman code - optimality - Lempel Ziv code - optimality for stationary ergodic sources.

Module 3 (14 Hours)
Channel Coding: BSC and BEC channel models - Channel capacity - Shannon's channel coding theorem - existence of capacity achieving codes for BEC, Fano-Elias Inequality.

Module 4 (14 Hours)
Cryptography: Information theoretic security - Perfect secrecy - Shannon's theorem - perfectly secret codes - Introduction to computational security and pseudo random sources.

References:

1. T. M. Cover and J. A. Thomas, Elements of Information Theory, Addison Wesley, 1999.
2. D. J. Mackay, Information Theory, Inference and Learning Algorithms. Cambridge University Press, 2002.
3. H. Delfs and H. Knebl, Introduction to Cryptography(2/e), Springer, 2010.

CS4025 GRAPH THEORY AND COMBINATORICS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
4	0	0	4

Total Hours: 56 Hrs

Module 1 (14 Hours)
Generating functions and applications: Power series expansion and generating functions, Catalan and Stirling numbers, solving recurrence equations using generating functions, Lambert series, Bell series and Dirichlet series, Applications.

Module 2 (14 Hours)
Existential Combinatorics: Ramsey theory, Ramsey theorem, Ramsey numbers, lower bound for R(k,k), Lovasz local lemma - bound on R(k,k) using Lovasz lemma, applications of local lemma.

Module 3 (14 Hours)
Matching theory: Bipartite matching, Konig's theorem, Hall's Matching Theorem, Network flow, Max flow min cut theorem, integrality, Ford Fulkerson method
Connectivity: Properties of 2 connected and 3 connected graphs, Menger's theorem, Applications

Module 4 (14 Hours)
Planar graphs and Colouring: Planar graphs, 5 color theorem, Brook's theorem, edge coloring, Vizing's theorem, list colouring, Thomassen's theorem.

References:

1. R. P. Grimaldi, Discrete and Combinatorial Mathematics, Addison Wesley, 1998.
2. R. P. Stanley. Enumerative Combinatorics, Cambridge University Press, 2001.
3. P. J. Cameron, Combinatorics: Topics, Techniques and Algorithms, Cambridge University Press, 1995.

CS4026 COMBINATORIAL ALGORITHMS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Formulate the primal and dual LP for a given optimization problem specification.

CO2: Design approximation algorithms using randomized rounding and primal-dual schema

CO3: Formulate matching and connectivity problems in graph theory as integer linear programs and establish duality relations between them.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Network Flows: Review of graph theory – spanning trees, shortest paths. Connectivity, Network Flows - Max flow min cut theorem, algorithms of Ford-Fulkerson, Edmond Karp, preflow-push algorithms.

Module 2 (10 (T) + 7(P) Hours)
Primal Dual Theory: Linear programming – Primal dual theory, LP-duality based algorithm design. Applications to Network flow and other combinatorial problems, Applications to graph theory - Konig's theorem, Halls theorem, Menger's theorem.

Module 3 (10 (T) + 7(P) Hours)
Matching Theory: Tutte's theorem, Primal dual algorithms – Hungarian algorithm, Edmond's maximum matching algorithm. Application to marriage problems, Hopcroft Karp algorithm.

Module 4 (12 (T) + 7(P) Hours)
Approximation: Primal Dual approximation algorithms for set cover, Maximum satisfiability, Steiner tree, multicut, Steiner forest, sparsest cut and k-medians.

References:

1. D. West, Graph Theory, Prentice Hall, 2002.
2. D. Jungnickel, Graphs Networks and Algorithms, Springer 2005.
3. U. Vazirani, Approximation Algorithms, Springer 2003.
4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, S. C. Stein, Introduction to Algorithms (4/e), McGraw Hill, 2010.

CS4027 TOPICS IN ALGORITHMS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1: Design Markov chain and Random walk models from problem specifications and perform steady state analysis

CO2: Compute the expected running time and tail bounds for Las Vegas algorithms

CO3: Derandomize algorithms using the method of conditional expectations

CO4: Derive performance lower bounds using methods from algorithmic information theory.

Total Hours: 56 Hrs

Module 1 (14 Hours)

Discrete Probability: Probability, Expectations, Tail Bounds, Chernoff Bound, Markov Chains. Random Walks Exponential Generating Functions, homogeneous and non-homogeneous of first and second degrees. Review of algorithm analysis.

Module 2 (14 Hours)

Randomized Algorithms, Moments and Deviations. Tail Inequalities. Randomized selection.
Las Vegas Algorithms. Monte Carlo Algorithms. Parallel and Distributed Algorithms. De-Randomization
Complexity: Probabilistic Complexity Classes

Module 3 (14 Hours)

Proof Theory. Examples of probabilistic algorithms. Probabilistic Method and Proofs, Proving that an algorithm is correct 'Almost sure'. Complexity analysis of probabilistic algorithms, Probabilistic Counting. Super recursive algorithms and inductive Turing machines

Module 4 (14 Hours)

Kolmogorv Complexity – Basic concepts. Models of Computation. Applications to analysis of algorithms. Lower bounds. Relation to Entropy. Kolmogorov complexity and universal probability. Godel's Incompleteness Theorem. Chatin’s Proof for Godel’s Theorem.

References:

1. R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995
2. C. H. Papadimitriou, Computational Complexity, Addison Wesley, 1994
3. Dexter C. Kozen, The Design and Analysis of Algorithms, Springer Verlag N.Y, 1992
4. Ronald Graham, Donald Knuth, Oren Patashnik (1989): Concrete Mathematics, Addison-Wesley, ISBN 0-201-14236-8
5. Current Literature

CS4028 QUANTUM COMPUTATION

Pre-requisite: Nil

ELECTIVE

L	T	P	C
4	0	0	4

Total Hours: 56 Hrs

Module 1 (14 Hours)
Review of Linear Algebra. The postulates of quantum mechanics. Review of Theory of Finite Dimensional Hilbert Spaces and Tensor Products

Module 2 (14 Hours)
Complexity classes. Models for Quantum Computation. Qubits. Single and multiple qubit gates. Quantum circuits. Bell states. Single qubit operations. Controlled operations and measurement. Universal quantum gates. Quantum Complexity classes and relationship with classical complexity classes

Module 3 (14 Hours)
Quantum Algorithms – Quantum search algorithm - geometric visualization and performance. Quantum search as a quantum simulation. Speeding up the solution of NP Complete problems. Quantum search as an unstructured database. Grover's and Shor's Algorithms.

Module 4 (14 Hours)
Introduction to Quantum Coding Theory. Quantum error correction. The Shor code. Discretization of errors, Independent error models, Degenerate Codes. The quantum Hamming bound. Constructing quantum codes – Classical linear codes, Shannon entropy and Von Neuman Entropy.

References:

1. Nielsen, Michael A., and Isaac L. Chuang, Quantum Computation and Quantum Information. Cambridge, UK, Cambridge University Press, September 2002
2. Gruska, J. Quantum Computing, McGraw Hill, 1999.
3. Halmos, P. R. Finite Dimensional Vector Spaces, Van Nostrand, 1958.
4. Peres, Asher. Quantum Theory: Concepts and Methods. New York, NY: Springer, 1993. ISBN: 9780792325499.

CS4029 TOPICS IN THEORY OF COMPUTATION

Pre-requisite: CS3001 Theory of Computation

ELECTIVE

L	T	P	C
4	0	0	4

Total Hours: 56 Hrs

Module 1 (14 Hours)

Recursion, The primitive recursive functions, Turing machines, Arithmetization, Coding functions , The normal form theorem, The basic equivalence and Church's thesis, Canonical coding of finite sets, Computable and computably enumerable sets, Diagonalization, Computably enumerable sets , Undecidable sets , Uniformity, Many-one reducibility, The recursion theorem, Proof for Godel's Incompleteness Theorem based on Recursion theorem.

Module 2 (14 Hours)

The arithmetical hierarchy, Computing levels in the arithmetical hierarchy , Relativized computation and Turing degrees, Turing reducibility , Limit computable sets, Incomparable degrees

Module 3 (14 Hours)

The priority method, Diagonalization, Turing incomparable sets , Undecidability , Constructivism, randomness and Kolmogorov complexity, Compressibility and randomness, Undecidability

Module 4 (14 Hours)

Scheme, programming and computability theory based on a term-rewriting, "substitution" model of computation by Scheme programs with side-effects; computation as algebraic manipulation: Scheme evaluation as algebraic manipulation and term rewriting theory.

References:

1. R. I. Soare, Recursively enumerable sets and degrees, Springer-Verlag, 1987
2. G. E. Sacks, Higher recursion theory, Springer Verlag, 1990.
3. M. Li and P. Vitányi, An introduction to Kolmogorov complexity and its applications, Springer-Verlag, 1993
4. Dexter C. Kozen, Automata and Computability, Springer-Verlag, Inc., New York, NY, 1997.
5. S. C. Kleene, Introduction to Metamathematics, Van Nostrand Co., Inc., Princeton, New Jersey, 1950.
6. MIT OpenCourseWare on Computability Theory of and with Scheme at http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-844-computability-theory-of-and-with-scheme-spring-2003/ accessed on 26/11/2010

CS4030 COMPUTATIONAL COMPLEXITY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
4	0	0	4

Course Outcomes:

CO1: Place a given computational problem to the appropriate complexity class

CO2: Establish connections between problems using reductions

CO3: Prove completeness of a given problem with respect to a given complexity class

CO4: Prove hardness of approximation using gap reductions

Total Hours: 56 Hrs

Module 1 (14 Hours)

Review of Complexity Classes, NP and NP Completeness, Space Complexity, Hierarchies, Circuit satisfiability, Savitch and Immerman theorems, Karp Lipton Theorem.

Module 2 (14 Hours)

Randomized Complexity classes, Adleman's theorem, Sipser Gacs theorem, Randomized Reductions, Counting Complexity, Permanent’s and Valiant’s Theorem

Module 3 (14 Hours)

Parallel complexity, P-completeness, Sup-liner space classes, Renegold's theorem, Polynomial hierarchy and Toda's theorem

Module 4 (14 Hours)

Arthur Merlin games, Graph Isomorphism problem, Goldwasser-Sipser theorem, Interactive Proofs, Shamir's theorem.

References:

1. S. Arora, B. Barak, Computational Complexity: A Modern Approach, Cambridge University Press, 2009.
2. Papadimtriou C. H.., Computational Complexity, Addison Wesley, First Edition, 1993
3. Motwani R, Randomized Algorithms, Cambridge University Press, 1995.
4. Vazirani V., Approximation Algorithms, Springer, First Edition, 2004.

CS4031 COMPUTATIOAL ALGEBRA

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Number Theory: Review of groups and rings and vector spaces, Euclid's algorithm, Structure of the ring Z_n Algorithms for computation in the ring Z_n - modular inversion, exponentiation, Chinese remaindering.

Module 2 (10 (T) + 7(P) Hours)
Finite fields: Structure theory of finite fields - Factorization of polynomials over finite fields - Berlekamp's algorithm, Cantor Zassenhaus algorithm, Fourier Transform algorithm for finite fields.

Module 3 (10 (T) + 7(P) Hours)
Primality Testing: Solovay Strassen test, Miller Rabin test, Agrawal, Kayal Saxena algorithm.

Module 4 (12 (T) + 7(P) Hours)
Applications: Euclid's algorithm for rational polynomial approximation and decoding BCH and RS codes. Applications to public key cryptography.

References:

1. V. Shoup, A computational Introduction to Number Theory and Algebra, Cambridge University Press, 2005.
2. H. Delfs and H. Knebl, Introduction to Cryptography, Springer, 1998.
3. J. von zur Gathen, Modern Computer Algebra, Cambridge University Press, 2003.
4. W. C. Huffman and V. Pless, Fundamentals of Error Correcting Codes, Cambridge University press, 2003.

CS4032 COMPUTER ARCHITECTURE

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1**:** Analyze the performance of a processor using standard benchmark suites

CO2: Define Instruction Level Parallelism(ILP) and design mechanisms on hardware and software level to exploit ILP

CO3: Design and optimize the performance of memory hierarchy for modern processors.

CO4: Define Thread Level Parallelism(TLP) and design techniques to exploit TLP using uniprocessors and multi processors.

CO5: Design cache coherence protocols and memory consistency models for multiprocessor environment

CO6: Define Data level parallelism (DLP) and design hardware to exploit DLP

Total Hours: 70 Hrs

Module 1 (8(T) + 7(P) Hours)

Fundamentals – Technology trend -performance measurement –Comparing and summarizing performance- quantitative principles of computer design –Amdahl's law- instruction set architectures – memory addressing- –type and size operand - encoding an instruction set - role of compilers - case study – MIPS 64 architecture – Review of pipelining - MIPS architecture

Module 2 (10(T) + 7(P) Hours)

Instruction level parallelism and its limits - dynamic scheduling —-dynamic hardware prediction - multiple issue processor – multiple issue with dynamic scheduling-hardware based speculation-limitation of ILP-Case study P6 micro-architecture Introduction to multicore processors,

Module 3 (16(T) + 12(P) Hours)

Multiprocessor and thread level parallelism- classification of parallel architecture-models of communication and memory architecture-Symmetric shared memory architecture-cache coherence protocols-distributed shared memory architecture-directory based cache coherence protocol- Memory consistency-relaxed consistency models multi threading- exploiting thread level parallelism multicore architecture, Memory hierarchy design - reducing cache misses and miss penalty, reducing hit time - main memory organization - virtual memory and its protection -. Memory issues in multicore processor based systems

-

Module 4 (8(T) + 2(P) Hours)

Storage Systems, Faults and reliability, Networks, Queuing, Design of storage systems – case studies

References

1. Hennesy J. L. & Pattersen D. A., Andrea C. Arpaci-Dusseau, Computer Architecture: A Quantitative approach, 4/e, Morgan Kaufman, 2007

2. Pattersen D. A. & Hennesy J. L., Computer Organisation and Design: The Hardware/ Software Interface, 3/e, Harcourt Asia Pte Ltd (Morgan Kaufman), Singapore

CS4033 DISTRIBUTED COMPUTING

Pre-requisite: CS2005 Data Structures and Algorithms

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10(T) + 7(P) Hours)

Characteristics of Distributed Systems, Distributed systems Versus Parallel systems, Models of distributed systems**,** Happened Before and Potential Causality Model, Models based on States, Logical clocks, Vector clocks, Verifying clock algorithms, Direct dependency clocks.

Module 2 (10(T) + 7(P) Hours)

Mutual exclusion using Time stamps, Distributed Mutual Exclusion (DME) using timestamps, token and Quorums, Centralized and distributed algorithms, proofs of correctness and complexity analysis. Drinking philosophers problem, Dining philosophers problem under heavy and light load conditions. Implementation and performance evaluation of DME algorithms.

Module 3 (10(T) + 7(P) Hours)

Leader election algorithms, Global state detection, Global predicates, Termination Detection, Control of distributed computation, disjunctive predicates. Performance evaluation of leader election algorithms on simulated environments.

Module 4 (12(T) + 7(P) Hours)

Self stabilization, knowledge and common knowledge, Distributed consensus, Consensus under Asynchrony and Synchrony, Check pointing for Recovery, R- Graphs

References:

1. Vijay K. Garg., Elements of Distributed Computing, Wiley & Sons, 2002
2. Sukumar Ghosh, Distributed Systems An Algorithmic Approach, Chapman & Hall, CRC Computer and Information Science Series, 2006.
3. Tanenbaum S, Distributed Operating Systems, Pearson Education.,2005
4. Coulouris G, Dollimore J. & Kindberg T., Distributed Systems Concepts And Design, 2/e, Addison Wesley 2004
5. Chow R. and Johnson T., Distributed Operating Systems and Algorithms, Addison Wesley, 2002

CS4034 MIDDLEWARE TECHNOLOGIES

Pre-requisite: CS4033 Distributed Computing

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Publish/Subscribe matching algorithm, event based systems, notification filtering mechanisms, Composite event processing, content based routing, content based models and matching, matching algorithms, distributed hash tables (DHT)

Module 2 (10 (T) + 7(P) Hours)
Distributed notification routing, content based routing algorithms, engineering event based systems, Accessing publish/subscribe functionality using APIs. Scoping, event based systems with scopes, notification mappings, transmission policies, implementation strategies for scoping.

Module 3 (10 (T) + 7(P) Hours)
Composite event detection, detection architectures, security, fault tolerance, congestion control, mobility, existing notification standards- JMS, DDS, HLA.

Module 4 (12 (T) + 7(P) Hours)
Topic based systems, Overlays, P2P systems, overlay routing, Case studies- REBECA, HERMES, Gryphon. Commercial systems- IBM Websphere MQ, TIBCO Rendezvous.

References:

1. Gero Muhl, Ludger Fiege, Peter R. Pietzuch, Distributed Event Based Systems. Springer, 2006
2. Chris Britton and Peter Bye, IT Architectures and Middleware. Pearson Education, (2/e), 2005
3. Yanlei Diao, and Michael J. Franklin, Query Processing for High-Volume XML Message Brokering. VLDB 2003.
4. Chee-Yong Chan, Minos Garofalakis and Rajeev Rastogi, RE-Tree: An Efficient Index Structure for Regular Expressions, VLDB 2002.
5. Peter R. Pietzuch, Brian Shand, Jean Bacon. A Framework for Event Composition in Distributed Systems, Proc. of the 4th Int. Conf. on Middleware (MW'03)

CS4035 COMPUTER SECURITY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Identify the confidentiality, integrity, availability requirements for a given information domain.

CO2: Analyze the efficiency and effectiveness of Asymmetric and Symmetric Key Cryptosystems.

CO3: Work with tools like Wireshark, Fiddler, Metasploit

CO4: Analyze basic security loopholes and develop defenses against few threats.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Operating system security - Access Control – MAC, DAC, RBAC. Formal models of security - BLP, Biba, Chinese Wall and Clark Wilson. Overview of SE Linux. Software vulnerabilities - Buffer and stack overflow, Phishing. Malware - Viruses, Worms and Trojans.

Module 2 (14 (T) + 7(P) Hours)
Network Security - Security at different layers – IPSec / SSL / PGP. Security problems in network domain - DoS, DDoS, ARP spoofing and session hijacking. DNS attacks and DNSSEC. Cross-site scripting XSS worm, SQL injection attacks. Intrusion Detection Systems (IDS). DDoS detection and prevention in a network.

Module 3 (9 (T) + 7(P) Hours)
Security in current domains – WEP - Wireless LAN security - Vulnerabilities - frame spoofing. Cellphone security - GSM and UMTS security. Mobile malware - bluetooth security.

Module 4 (9 (T) + 7(P) Hours)
Security in current applications – Security case studies of Online banking and Credit Card Payment Systems. Challenges in security for web services and clouds.

References:

1. Bernard Menezes, Network security and Cryptography, Cengage Learning India, 2010.
2. B A Forouzan and D Mukhopadyay, Cryptography and Network Security(2/e). Tata McGraw Hill, 2010
3. Dieter Gollmann, Computer Security, John Wiley and Sons Ltd., 2006.

CS4036 ADVANCED DATABASE MANAGEMENT SYSTEMS

Pre-requisite: CS3002 Database Management Systems

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Model, Design and develop concurrent, distributed and spatial database applications

CO2: Write reports, surveys and possibly publish on the advances in the database field in conferences/journals.

CO3: Query spatial databases using spatial query languages.

CO4: Port existing database applications into the cloud database environment.

CO5: Deploy efficient database solutions using free and open software.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Distributed database concepts - overview of client - server architecture and its relationship to distributed databases, Concurrency control Heterogeneity issues, Persistent Programming Languages, Object Identity and its implementation, Clustering, Indexing, Client Server Object Bases, Cache Coherence.

Module 2 (10 (T) + 7(P) Hours)

Parallel Databases: Parallel Architectures, performance measures, shared nothing/shared disk/shared memory based architectures, Data partitioning, Intra-operator parallelism, Pipelining, Scheduling, Load balancing, Query processing- Index based, Query optimization: cost estimation, Query optimization: algorithms, Online query processing and optimization, XML, DTD, XPath, XML indexing, Adaptive query processing

Module 3 (10 (T) + 7(P) Hours)

Advanced Transaction Models: Savepoints, Sagas, Nested Transactions, Multi Level Transactions. Recovery: Multi-level recovery, Shared disk systems, Distributed systems 2PC, 3PC, replication and hot spares, Data storage, security and privacy- Multidimensional K- Anonymity, Data stream management.

Module 4 (12 (T) + 7(P) Hours)

Models of Spatial Data: Conceptual Data Models for spatial databases (e.g. pictogram enhanced ERDs), Logical data models for spatial databases: raster model (map algebra), vector model, Spatial query languages, Need for spatial operators and relations, SQL3 and ADT. Spatial operators, OGIS queries

References:

1. Avi Silberschatz, Hank Korth, and S. Sudarshan. Database System Concepts, (5/e), McGraw Hill, 2005
2. S. Shekhar and S. Chawla. Spatial Databases: A Tour, Prentice Hall, 2003.
3. Ralf Hartmut Guting, Markus Schneider, Moving Objects Databases Morgan Kaufman, 2005.
4. R. Elmasri and S. Navathe, Fundamentals of Database Systems, Benjamin- Cummings ,(5/e), 2007

CS4037 CLOUD COMPUTING

Pre-requisite: CS4033 Distributed Computing

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
New Computing Paradigms & Services: Cloud computing , Edge computing , Grid computing , Utility computing , Cloud Computing Architectural Framework, Cloud Deployment Models, Virtualization in Cloud Computing, Parallelization in Cloud Computing, Security for Cloud Computing, Cloud Economics , Metering of services.

Module 2 (10 (T) + 7(P) Hours)
Cloud Service Models: Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Service Oriented Architecture (SoA), Elastic Computing, On Demand Computing, Cloud Architecture, Introduction to virtualization.

Module 3 (10 (T) + 7(P) Hours)
Types of Virtualization, Grid technology , Browser as a platform, Web 2.0, Autonomic Systems, Cloud Computing Operating System, Deployment of applications on the cloud, Case studies- Xen, VMware, Eucalyptus, Amazon EC2.

Module 4 (12 (T) + 7(P) Hours)
Introduction to Map Reduce, Information retrieval through Map Reduce, Hadoop File System, GFS, Page Ranking using Map Reduce, Security threats and solutions in clouds, mobile cloud computing, Case studies- Ajax, Hadoop.

References:

1. Tom White, Hadoop: The Definitive Guide, O'Reilly Media, 2009
2. Jason Venner, Pro Hadoop, Apress, 2009
3. Timothy Chou , Introduction to cloud computing & Business, Active Book Press, 2010
4. Current literature- Journal & conference papers

CS4038 DATA MINING

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Identify the data mining methodology for solving a given data set.

CO2: Model methodologies for cleaning/pre-processing the data set based on classification, regression and other data cleaning techniques.

CO3. Evolve solutions with *Interestingness* from the pre-processed data by applying data mining models on it.

CO4. Solve and find useful, previously unseen results for any given data set without using conventional SQL queries.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Introduction to data mining-challenges and tasks Data preprocessing data analysis, measures of similarity and dissimilarity, Data visualization –concepts and techniques

Module 2 (10 (T) + 7(P) Hours)

Classification- decision tree-performance evaluation of the classifier, comparison of different classifiers, Rule based classifier, Nearest-neighbor classifiers-Bayesian classifiers-support vector machines, Class imbalance problem

Module 3 (10 (T) + 7(P) Hours)

Association analysis –frequent item generation rule generation, evaluation of association patterns

Module 4 (12 (T) + 7(P) Hours)

Cluster analysis,-types of clusters, K means algorithm, cluster evaluation, application of data mining to web mining and Bioinformatics

References:

1. Pang-Ning Tan,Michael Steinbach and Vipin Kumar , *Introduction to Data Mining*, Pearson Education 2006.
2. Han and Kamber, *Data Mining: Concepts and Techniques* (2e), Morgan Kaufmann, 2005.

CS4039 MULTI AGENT SYSTEMS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Define an agent and model agent-based solutions for distributed problem solving.

CO2: Use software tools for modeling multi-agent based solutions.

CO3: Design of protocols for agent-agent communications.

CO4: Design of different negotiation models: game theoretical, heuristic approach and argumentation based approach.

CO5: Formulate solutions for distributed decision-making and design criteria for the evaluation of decisions.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Introduction to agent and multi-agent systems, different types of agents, abstract architecture, distributed problem solving, application areas, Software tools for modeling Multi-Agent Systems

Module 2 (10 (T) + 7(P) Hours)
Agent communication, communication languages KQML and FIPA ACL Communication policies and protocols, Protocol Modeling

Module 3 (10 (T) + 7(P) Hours)
Negotiation in multi-agent- agent environment, game theoretical model , heuristic approach, argumentation based approach

Module 4 (12 (T) + 7(P) Hours)
Distributed decision making –evaluation criteria -Social welfare, Pareto Efficiency, Individual Rational, Stability, Application of multiagent systems in complex distributed problem solving, Modeling distributed multi-agent systems.

References:

1. M. Wooldrige, *An Introduction to multi-agent systems*, Wiley, 2009.
2. R. Norvig, *Artificial Intelligence: A modern approach*, Prentice Hall, 2010.

CS4040 BIOINFORMATICS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Define and describe the importance of biomolecules like DNA, RNA and Protein

CO2. Abstract and formalize bioinformatics problems

CO3. Apply various algorithm design and analysis techniques to solve bioinformatics problems

CO4. Implementation of various bioinformatics algorithms

CO5. Apply different bioinformatics tools and databases

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Molecular biology primer, gene structure and information content, Bioinformatics tools and databases, genomic information content, Sequence Alignment, Algorithms for global and local alignments, Scoring matrices, Dynamic Programming algorithms.

Module 2 (10 (T) + 7(P) Hours)

Introduction to Bio-programming languages, Restriction Mapping and Motif finding, Gene Prediction, Molecular Phylogenetics, Phylogenetic trees, Algorithms for Phylogenetic Tree construction.

Module 3 (10 (T) + 7(P) Hours)

Combinatorial pattern matching, Repeat finding, Keyword Trees, Suffix Trees, Heuristic similarity search algorithms, Approximate pattern matching.

Module 4 (12 (T) + 7(P) Hours)

Microarrays, Gene expression, Algorithms for Analyzing Gene Expression data, Protein and RNA structure prediction, Algorithms for structure prediction. Emerging trends in bioinformatics algorithms and databases.

References:

1. Neil C Jones and Pavel A Pevzner, An Introduction to Bioinformatics Algorithms, MIT Press, 2004.
2. David W Mount, Bioinformatics- Sequence and Genome Analysis, (2/e), Cold Spring Harbor Laboratory Press, New York, 2004.
3. D. E. Krane and M. L. Raymer, Fundamental Concepts of Bioinformatics, Pearson Education, 2003.
4. T. K. Attwood and D. J. Parry-Smith, Introduction to Bioinformatics, Pearson Education, 2003.
5. Current Literature.

CS4041 NATURAL LANGUAGE PROCESSING

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Define and describe the various challenges associated with natural language understanding such as syntactic ambiguity, semantic ambiguity, discourse analysis and pragmatics.

CO2. Relate fundamental mathematical principles of probability theory, statistics and linear algebra and basic computer science principles like dynamic programming and parsing.

CO3. Explain and apply various important tasks in natural language processing such as language modeling, information extraction, named entity recognition, information retrieval, text classification, word sense disambiguation, automatic question answering and text summarization.

CO4. Use natural language processing (NLP) tools and libraries (such as python, nltk) and develop softwares for various NLP tasks such as tagging, parsing and text classification.

CO5. Recall fundamental grammatical structure of natural languages, with a special emphasis on English grammar.

Total Hours: 70 Hrs

Module 1 (10(T)+7(P) Hours)

Introduction to Natural Language Processing, Different Levels of language analysis, Representation and understanding, Linguistic background. Grammars and parsing, Top down and Bottom up parsers.

Module 2 (10(T)+7(P) Hours)

Transition Network Grammars, Feature systems and augmented grammars, Morphological analysis and the lexicon, Parsing with features, Augmented Transition Networks.

Module 3 (10(T)+7(P) Hours)

Grammars for natural language, Movement phenomenon in language, Handling questions in context free grammars, Hold mechanisms in ATNs, Gap threading, Human preferences in parsing, Shift reduce parsers, Deterministic parsers, Statistical methods for Ambiguity resolution

Module 4 (12(T)+7(P) Hours)

Semantic Interpretation, word senses and ambiguity, Basic logical form language, Encoding ambiguity in logical from, Thematic roles, Linking syntax and semantics, Information Retrieval, Recent trends in NLP.

References:

1. James Allen, Natural Language Understanding (2/e), Pearson Education, 2003
2. T Siddiqui and U S Tiwary, Natural Language Processing and Information Retrieval, Oxford University Press, 2008
3. D Juraffsky and J H Martin, Speech and Language Processing, Pearson Education, 2000

CS4042 WEB PROGRAMMING

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Develop web solutions using appropriate technologies.

CO2 : Analyze the complexity of a problem and propose a web based solution.

CO3 : Design and develop of web applications in languages like PHP and RoR

CO4 : Describe application server platforms like WAMP and Instant Rails.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Internet and its architecture, Client Server Networking - Creating an Internet Client, Application Protocols and http, Presentation aspects html, CSS and Java script, Creating a web server, Serving Dynamic Content- CGI – overview of technologies like PHP – applets – JSP. Implementation examples.

Module 2 (10 (T) + 7(P) Hours)

Web server architecture, Programming threads in C, Shared memory synchronization, Performance measurement and workload models. Comparison using existing benchmarks.

Module 3 (10 (T) + 7(P) Hours)

Web development frameworks – Detailed study of one open source web framework - Ruby Scripting, Ruby on rails – Design, Implementation and Maintenance aspects.

Module 4 (12 (T) + 7(P) Hours)

Service Oriented Architecture – SOAP. Web 2.0 technologies. – AJAX. Development using Web2.0 technologies. Introduction to semantic web.

References:

1. Dave Thomas, with Chad Fowler and Andy Hunt. Programming Ruby: The Pragmatic Programmer's Guide (3/e), Pragmatic Programmers, May 2008.

2. Balachander Krishnamurthy and Jennifer Rexford. Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement (1/e), Addison Wesley Professional, 2001

CS4043 IMAGE PROCESSING

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Explain and Apply the fundamental concepts of image processing.

CO2. Design solutions for simple and complex image processing problems using the various spatial domain and transform domain techniques.

CO3. Apply various filtering techniques and segmentation methods for image enhancement and image segmentation.

CO4. Explain and apply the apt noise removal techniques for images depending the application and explain the image reconstruction techniques.

CO5. Build solutions for various image processing problems in different domains like medical imaging, satellite imaging etc..

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Fundamentals of Image processing: Digital image representation, Elements of Digital image processing systems, Image model, Sampling and Quantization, Basic relations between pixels.

Image transforms: One dimensional Fourier transform, Two dimensional Fourier transform, Properties of two dimensional Fourier transform. Walsh transform, Hadamard transform, Discrete cosine transform, Haar transform, Slant transform.

Module 2 (10 (T) + 7(P) Hours)

Image enhancement techniques: Spatial domain methods, Frequency domain methods, Intensity transform, Histogram processing, Image subtraction, Image averaging, Smoothing filters, Sharpening filters, Spatial masks from frequency domain.

Module 3 (10 (T) + 7(P) Hours)

Image Segmentation: Thresholding: Different types of thresholding methods, Point detection, Edge detection: Different types of edge operators, Line detection, Edge linking and boundary detection, Region growing, Region splitting, Region Merging.

Module 4 (12 (T) + 7(P) Hours)

Image Data Compression: Fundamentals, Compression models, Error free compression, Lossy Compression, Image compression standards.

Applications of Image Processing: Medical imaging, Robot vision, Character recognition, Remote Sensing.

References:

1. R.C.Gonzalez and R.E.Woods, . Digital Image Processing, Addison-Wesley Publishing Company, 2007.

2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, (2/e), PWS Publishing, 1999

CS4044 PATTERN RECOGNITION

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Differentiate the kinds of Pattern Recognition tasks such as supervised, unsupervised learning, classification and regression.

CO2. Relate fundamental mathematical principles of probability theory, linear algebra and optimization to solve pattern recognition tasks.

CO3. Describe linear classifiers like logistic regression, least squares classifier and perceptron, and non-linear classifiers like SVM and artificial neural networks.

CO4. Use classification tools and libraries (such as weka, iPython notebook) for solving real world pattern recognition problems.

CO5. Model various socially important problems related to domains such as health care, weather forecasting, object recognition, anomaly detection etc., as pattern recognition problems, by extraction of proper features, selection of right algorithms, and evaluation of results.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Introduction: Machine Perception , Pattern Recognition Systems, The Design Cycle, Learning and Adaptation.
Baye's Decision Theory: Bayes Decision Theory, Minimum Error rate Classification, Classifiers, Discriminant functions and Decision Surfaces, Normal Density, Discriminant functions for the Normal Density, Bayes Decision Theory for Discrete features

Module 2 (10 (T) + 7(P) Hours)
Maximum Likelihood and Bayesian Parameter Estimation : Maximum Likelihood Estimation, Bayesian Estimation, Bayesian Parameter Estimation, Gaussian Case and General Theory.
Non Parametric Techniques: Density Estimation, Parzen Windows , K- Nearest Neighbor Estimation,NN rule, Metrics and NN Classification, Fuzzy Classification

Module 3 (10 (T) + 7(P) Hours)
Linear Descriminant Functions : Linear Discriminant Functions and Decision Surfaces, Generalized Discriminant Functions, The two-category linearly separable case , Minimizing the perceptron criterion function, relaxation procedures, non- separable behavior, Minimum Squared- Error procedures.

Module 4 (12 (T) + 7(P) Hours)
Multi Layer Neural Networks : Feed-forward Operation, Classification, Back – propagation Algorithm, Error Surfaces, Back-propagation as Feature mapping.

References:

1. R. O. Duda, P. E. Hart and D. G. Stork, Pattern *Classification* , John-Wiley, 2004
2. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles,* by Tou and Gonzalez, Wiley, 1974.

CS4045 MEDICAL IMAGE PROCESSING

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Introduction to digital image processing: images, image quality, basic operations.
Radiography: Introduction, X-rays, interaction with matter, detectors, dual energy imaging, quality clinical use, biologic effect and safety, Fourier Slice Theorem Basics.

Module 2 (10 (T) + 7(P) Hours)
X-ray Computed tomography: Introduction, X-ray detectors in CT, imaging, cardiac CT, image quality, clinical use, biologic effects and safety.
Magnetic resonance imaging: Introduction, physics of transmitted signal, interaction with tissue, signal detection and detector, imaging. Biologic effects and safety

Module 3 (10 (T) + 7(P) Hours)
Nuclear imaging, Introduction, radionuclides, interaction of Gama-photons and particles with matter, data acquisition, imaging, image quality, equipment, clinical use, biologic effects and safety
Ultrasound imaging: Physics of acoustic waves, generation and detection of ultrasound, grayscale imaging, Doppler imaging, image quality, equipment, clinical use, biologic effects and safety.

Module 4 (12 (T) + 7(P) Hours)
Medical image analysis: Manual and automated analysis, computation strategies for automated medical image analysis, pixel classification.

References:

1. Paul Suetens, *Fundamentals of medical imaging*, Cambridge University Press, 2009
2. Bushberg, J. A. et al. *The Essential Physics of Medical Imaging (2e)*, L. Williams and Wilkins, 2002

CS4046 COMPUTER VISION

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Introduction and overview, pinhole cameras, radiometry terminology. Sources, shadows and shading: Local shading models- point, line and area sources; photometric stereo. Color: Physics of color; human color perception, Representing color; A model for image color; surface color from image color.

Module 2 (10 (T) + 7(P) Hours)
Linear filters: Linear filters and convolution; shift invariant linear systems- discrete convolution, continuous convolution, edge effects in discrete convolution; Spatial frequency and fourier transforms; Sampling and aliasing; filters as templates; Normalized correlations and finding patterns. Edge detection: Noise; estimating derivatives; detecting edges. Texture: Representing texture; Analysis using oriented pyramid; Applications; Shape from texture. The geometry and views: Two views.

Module 3 (10 (T) + 7(P) Hours)
Stereopsis: Reconstruction; human stereo; Binocular fusion; using color camera.

Module 4 (12 (T) + 7(P) Hours)
Segmentation by clustering: Human vision, applications, segmentation by graph theoretic clustering. Segmentation by fitting a model, Hough transform; fitting lines, fitting curves;

References:

1. David A Forsynth and Jean Ponce, Computer Vision- A modern approach, Pearson education series, 2003.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle , Digital image processing and computer vision, Cengage learning, 2008.
3. Schalkoff R. J., Digital Image Processing and Computer Vision, John Wiley, 2004.

CS4047 COMPUTER GRAPHICS

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Develop techniques for 3D modelling, image synthesis and rendering.

CO2 : Design lighting and shading techniques

CO3 : Devise techniques to produce 3D models using graphical software.

Total Hours: 70Hrs

Module 1 (10 (T) + 7(P) Hours)

Graphics Pipeline - overview of vertex processing, primitive generation, transformations and projections, clipping, rasterisation, fragment processing - Graphics Hardware - overview of GPU architecture, how GPUs SIMD architecture suits computer graphics.

Module 2 (10 (T) + 7(P) Hours)

Coordinate Systems - representations, homogenous coordinates, object, camera, world, and screen coordinate system, changing coordinate systems. Transformations - affine transformations, translation, rotation, scaling in homogenous coordinates, matrix representations, cumulation of transformations. Viewing and Projections - orthographic and perspective projection, camera positioning, Hidden Surface Removal - its importance in rendering, z buffer algorithm, clipping, culling, Data Structures for efficient implementation of the transformations and projections.

Module 3 (10 (T) + 7(P) Hours)

Lighting and Shading - light sources, normal computation, reflection models, flat and smooth shading , Introduction to Textures and Mapping - Rendering Techniques - slicing, volume rendering, iso-surface extraction, ray casting, multi resolution representations for large data rendering. Data Structures for efficient implementation.

Module 4 (12 (T) + 7(P) Hours)

Geometric Modelling - Data structures - tree representations, hierarchical models, scene graphs - particle systems and representations - introduction to modeling and solving dynamics based on physics, Introduction to Curves Surfaces (Bezier, splines) and Meshes - structured and unstructured.

<u>References:</u>

1. E. S. Angel, Interactive Computer Graphics, A top-down approach with OpenGL, (5e), Pearson Education, 2009..
2. D. Hearn and M. P. Baker, Computer Graphisc with OpenGL, Prentice Hall, 2003, (3/e), Prentice Hall, 2003.

CS4048: TOPICS IN COMPILERS

Prerequisite: CS 3005 Compiler Design

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1: Attribute grammars (10(T) + 7(P) hours)
Analysis, use, tests, circularity. Issues in type systems.

Module 2: Analysis and Optimizations (10(T)+7(P) hours)
Advanced topics in Data Flow, Control Flow and Dependency analysis, Loop optimizations – invariant code motion, elimination of partial redundancy, Experimental platforms – SUIF.

Module 3: ILP Compilation (11(T) + 7(P) hours)
Issues in compilation for ILP based processors. Effect of VLIW, Speculative, Predicated instructions, multithreaded processors.

Module 4: Dynamic Compilation (11(T)+7(P) hours)
Introduction, methods, case studies, implementation, software tools.

References:

1. ACM SIGPLAN.
2. ACM Transactions on Programming languages and Systems.
3. STEVEN MUCHNICK. *Advanced Compiler Design Implementation*, Morgan Kauffmann Publishers, 1997
4. Aho A.V, Lam M.S, Sethi R and Ullman J. D, Compilers – Principles, Techniques and Tools, Pearson, 2007.

CS4049 ADVANCED COMPUTER NETWORKS

Pre-requisite: CS3006 Computer Networks ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1. Define and describe Physical Layer and MAC layer protocols for any type of computer networking cases like LAN,MAN,and WAN and also for Wireless Networks.

CO2. Design and specify IPv4 and IPv6 classes and Routing protocols like OSPF and BGP

CO3. Specify the Transport Layer protocols particularly TCP and SCTP and studying in depth of congestion control and avoidance in TCP

CO4. Specify DNS and Network Security and firewall and Web server using the specification of Web 2.0

CO5. Specify network design and protocol design using open source softwares like NS2, Wireshark and CISCOs Packet tracer.

.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours): Introduction- Internet design philosophy, layering and end to end design principle. MAC protocols for high-speed LANS, MANs, wireless LANs and mobile networks, VLAN. Fast access technologies.

Module 2 (10 (T) + 7(P) Hours): IPv6: Why IPv6, basic protocol, extensions and options, support for QoS, security, neighbour discovery, auto-configuration, routing. Changes to other protocols. Application Programming Interface for IPv6, 6bone. IP Multicasting, wide area multicasting, reliable multicast. Routing layer issues, ISPs and peering, BGP, IGP, Traffic Engineering, Routing mechanisms: Queue management, packet scheduling. MPLS, VPNs

Module 3 (10 (T) + 7(P) Hours): TCP extensions for high-speed networks, transaction-oriented applications. New options in TCP, TCP performance issues over wireless networks, SCTP, DCCP.

Module 4 (12 (T) + 7(P) Hours): DNS issues, other naming mechanisms, overlay networks, p2p networks, web server systems, web 2.0, Internet traffic modelling, Internet measurements. Security – Firewalls, Unified threat Management System, Network Access Control.

References:

1. Adrian Farrel, The Internet and its protocols a comparative approach, Elsevier, 2005
2. M. Gonsalves and K. Niles._IPv6 Networks, McGraw Hill, 1998.
3. W. R. Stevens, TCP/IP Illustrated, Volume 1: The protocols, Addison Wesley, 1994.
4. G. R. Wright, TCP/IP Illustrated, Volume 2: The Implementation, Addison Wesley, 1995.
5. W. R. Stevens, TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the Unix Domain Protocols, Addison Wesley, 1996.
6. Articles in various journals and conference proceedings.
7. RFCs and Internet Drafts, available from Internet Engineering Task Force.

CS4050 DESIGN AND ANALYSIS OF ALGORITHMS

Pre-requisite: CS2005 Data Structures & Algorithms

ELECTIVE

L	T	P	C
3	0	2	4

Course Outcomes:

CO1: Analyzing the amortized time complexity of a given algorithm and data structure operations

CO2: Decide the appropriate design methodology for a given problem from among the paradigms of Divide and Conquer, Dynamic Programming, Greedy, Branch and Bound.

CO3: Prove the NP completeness of a given problem by using the technique of many-one reductions.

CO4: Compute the expected running time of a given randomized algorithm using probabilistic models.

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)

Analysis: RAM model - big Oh - big Omega – Asymptotic Analysis, recurrence relations, probabilistic analysis - linearity of expectations - worst and average case analysis of sorting algorithms, binary search - hashing algorithms - lower bound proofs for the above problems - amortized analysis - aggregate - accounting and potential methods - analysis of Knuth-Morris-Pratt algorithm - amortized weight balanced trees

Module 2 (10 (T) + 7(P) Hours)

Problem Solving, Classical Algorithm paradigms,: divide and conquer - Strassen's algorithm, O(n) median finding algorithm - dynamic programming - matrix chain multiplication - optimal polygon triangulation - optimal binary search trees - Floyd-Warshall algorithm - CYK algorithm - greedy - Huffman coding - Knapsack, Kruskal's and Prim's algorithms for MST - backtracking - branch and bound - traveling salesman problem - matroids and theoretical foundations of greedy algorithms

Module 3 (10 (T) + 7(P) Hours)

Complexity: complexity classes - P, NP, Co-NP, NP-Hard and NP-complete problems - cook's theorem- NP-completeness reductions for clique - vertex cover - subset sum - hamiltonian cycle - TSP - integer programming - approximation algorithms - vertex cover - TSP - set covering and subset sum

Module 4 (12 (T) + 7(P) Hours)

Probabilistic algorithms: pseudo random number generation methods - Monte Carlo algorithms - probabilistic counting - verifying matrix multiplication - primality testing - Miller Rabin test - integer factorization - Pollard’s rho heuristic - amplification of stochastic advantage - applications to cryptography - interactive proof systems - les vegas algorithms - randomized selection and sorting - randomized solution for eight queen problem - universal hashing - Dixon’s integer factorization algorithm

<u>References:</u>

1. Cormen T.H., Leiserson C.E, Rivest R.L. and Stein C, Introduction to Algorithms, Prentice Hall India, 3/e, 2010
2. Motwani R and Raghavan P., Randomized Algorithms, Cambridge University Press, 2001
3. Anany Levitin, Introduction to the Design & Analysis of Algorithms, Pearson Education. 2003
4. Basse S., Computer Algorithms: Introduction to Design And Analysis, Addison Wesley.
5. Manber U., Introduction to Algorithms: A Creative Approach, Addison Wesley
6. Aho A. V., Hopcroft J. E. & Ullman J. D., The Design And Analysis of Computer Algorithms, Addison Wesley

CS4051 CODING THEORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Linear Codes: Review of linear algebra - Linear codes and syndrome decoding. Generator and parity check matrices. Hamming geometry and code performance. Hamming codes. Error correction and concept of hamming distance.

Module 2 (10 (T) + 7(P) Hours)
Cyclic codes: BCH codes, Reed-Solomon codes – Polynomial time decoding. Shift register encoders for cyclic codes. Cyclic hamming codes. Decoding BCH – key equation and algorithms. Berlekamp's Iterative decoding Algorithm.

Module 3 (10 (T) + 7(P) Hours)
Convolutional codes : Viterbi decoding. Concept of forward error correction. State diagram, trellises. Concept of space time codes. Space Time Trellis codes. Path enumerators and proof of error bounds. Applications to wireless communication.

Module 4 (12 (T) + 7(P) Hours)
Codes on Graphs: Concept of girth and minimum distance in graph theoretic codes. Expander Graphs and Codes – linear time decoding. Basic expander based construction of list decodable codes. Sipser Spielman algorithm. Bounding results.

References:

1. R. Johannesson and K. Sh. Zigangirov, Fundamentals of Convolutional Coding, Wiley-IEEE Press, 1999.
2. W. C. Huffman and V. Pless, Fundamentals of error correcting codes, Cambridge University Press, 2003.
3. van Lint J. H. An Introduction to Coding Theory, (2/e). New York: Springer-Verlag, 1992.
4. R.J. McEliece, The Theory of Information and Coding, Addison Wesley, 1997.

CS4052 LOGIC FOR COMPUTER SCIENCE

Pre-requisite: Nil

ELECTIVE

L	T	P	C
3	0	2	4

Total Hours: 70 Hrs

Module 1 (10 (T) + 7(P) Hours)
Propositional logic, syntax of propositional logic, semantics of propositional logic, truth tables and tautologies, tableaus, soundness theorem, finished sets, completeness theorem.

Module 2 (10 (T) + 7(P) Hours)
Predicate logic, syntax of predicate logic, free and bound variables, semantics of predicate logic, graphs, tableaus, soundness theorem, finished sets, completeness theorem, equivalence relations, order relations, set theory.

Module 3 (10 (T) + 7(P) Hours)
Linear time Temporal Logic(LTL), syntax of LTL, semantics of LTL, Buchi Automata, Buchi recognizable languages and their properties, Automata theoretic methods, Vardi-Wolper Construction, Satisfiability problem of LTL, Model checking problem of LTL.

Module 4 (12 (T) + 7(P) Hours)
Software Verification: Introduction to Tools used for software verification - SPIN and SMV, Method of verification by the tools.

References:

1. Jerome Keisler and H. Joel Robbin, Mathematical Logic and Computability, McGraw-Hill International Editions, 1996
2. Papadimitriou. C. H., Computational Complexity, Addison Wesley, 1994
3. Gallier, J. H., Logic for Computer Science: Foundations of Automatic Theorem Proving, Harper and Row, 1986.

CS3091 COMPILER LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Design and development of a compiler

CO2. Analyze large systems.

CO3. Apply tools like Lex/Flex and Yacc/Bison for lexical analysis and syntax analysis.

CO4. Write large programs.

CO5. Synthesize large systems.

Total Hours: 56 Hrs

Theory (14 Hours) Practical (42 Hours)

Module 1 (2 (T) + 6(P) Hours)
Generation of lexical analyzer using tools such as LEX

Module 2 (6 (T) + 14(P) Hours)
Generation of parser using tools such as YACC. Creation of Abstract Syntax Tree

Module 3 (3 (T) + 10(P) Hours)
Creation of Symbol tables. Semantic Analysis.

Module 4 (3 (T) + 12(P) Hours)
Generation of target code.

References:

1.W. Appel, Modern Compiler Implementation in C , Cambridge University Press, 1998.
2. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, Compilers- Principles, Techniques & Tools (2/e), Pearson Education, 2007.

CS3092 OPERATING SYSTEMS LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1: Implement a primitive file system for a given design.

CO2: Implement multiprogramming support for a given specification.

CO3: Implement shell interface of a given specification.

CO4: Implement a given system call interface specification.

CO5: Use available Linux primitives to design and develop a system program for problems involving multiprogramming, IPC, shared memory.

Total Hours: 56Hrs

Theory (14 Hours)

Unix system programming fundamentals and system calls.

Practical (42 Hours)

Linux shell programming, Inter process communication-Pipes, semaphores, Shared memory and Message passing Loading executable programs into memory and execute System Call implementation-read(), write(), open () and close()

Multiprogramming-Memory management- Implementation of Fork(), Wait(), Exec() and Exit() System calls

Support for software TLB- TLB implementation – implementation of LRU replacement algorithm

File system implementation-demand paging - page fault exception – page replacement policy

Implementation of Synchronization primitives -Semaphore, Locks and Conditional Variables

Build Networking facilities - Mailbox

References:

1. Gary J. Nutt, Operating Systems, Pearson Education, 3/e, 2004.
2. Daniel P Bovet , Marco Cesati , Understanding the Linux Kernel, O'Reilly Media, (3/e), 2005
3. Course Web page

CS3093 NETWORKS LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Describe the basic socket programming functions for STREAM and DGRAM sockets.
CO2. Design and implement simple client server programs using STREAM and.DGRAM sockets.
CO3. Design and implement application in which server can accept multiple connection and do concurrent servicing of clients using either, fork, select/poll or threads.
CO4. Simulate network topology and analyze its performance using tools like NS2.
CO5. Use real time traffic analyzer tools like wireshark etc and other network utility tools.

Total Hours: 56 Hrs

Theory (14 Hours): Introduction, Overview of Unix Programming Environment, Unix Programing Tools, Introduction to Computer Networking and TCP/IP, Introduction to Socket Programming, TCP Sockets and Concurrent Servers, Threads, I/O Multiplexing and Socket Options, UDP Sockets and Name and Address Conversions, Daemon Processses and Inetd Superserver, Advanced I/O and Timeouts, Non-blocking Sockets, Unix Domain Sockets, Broadcasting, Multicasting, Advanced UDP Sockets, Ioctl Operations.
Introduction to open source firewall packages. Introduction to network emulators and simulators.

Practical (42 Hours)

Experiment 1: Implementation of basic Client Server program using TCP Socket (Eg. Day time server and clent).
Experiment 2: Implementation of basic Client Server program using UDP Socket.
Experiment 3: Implementing a program with TCP Server and UDP Client.
Experiment 4: Implementation of TCP Client Server program with concurrent connection from clients.
Experiment 5: Implementing fully concurrent application with a TCP server acting as a directory server and client programs allowing concurrent connection and message transfer (Eg. Chat sytem).
Experiment 6: Fully decentralized application like a Peer to Peer system. This program is to implement without a designated Sever as in the case of experiment 5.
Experiment 7: Experiments with open source firewall/proxy packages like iptables,ufw, squid etc.
Experiment 8: Experiments with Emulator like Netkit, Emulab etc.
Experiment 9: Experiments with Simulator like NS2, NCTU NS etc.

References:

1. W. Rıchard Stevens, Unix Network Programming – Networking APIs: Sockets and XTI Volume 1, 2nd Edition, Pearson Education, 2004.
2. W. Rıchard Stevens, Unix Network Programming – Interprocess Communications Volume 2, 2nd Edition, Pearson Education, 2004.
3. Warren W. Gay, Linux Socket Programming by Example, 1st Edition, Que Press, 2000.
4. Brian Hall, Beej's Guide to Network Programming, http://beej.us/guide/bgnet/
5. Elliotte Rusty Harold, Java Network Programming, 3rd Edition, O'Reilly, 2004.
6. Douglas C. Schmidt, and Stephen D. Huston, C++ Network Programming, Volume 2, Addison-wesley, 2003

CS3094 PROGRAMMING LANGUAGES LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)
Functional programming foundations review.
Practical (42 Hours)

Module 1 (5 (T) + 12(P) Hours)
Introduction to functional programming. Interpreter for the language of untyped arithmetic expressions.

Module 2 (3 (T) + 12(P) Hours)
Interpreter for the language of Untyped Lambda Calculus

Module 3 (3 (T) + 9(P) Hours)
Interpreter for the language of Typed arithmetic expressions.

Module 4 (3 (T) + 9(P) Hours)
Interpreter for Simply Typed Lambda Calculus and its extensions.

References:

1. Benjamin C. Pierce, Types and Programming Languages , MIT Press, 2002.

CS3095 DATABASE MANAGEMENT SYSTEMS LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Course Outcomes:

CO1. Design of Entity-Relationship model for real world problems and development using workbenches
CO2. Design and normalization of database based on ER model and functional dependencies
CO3. Create and query a database
CO4. Develop web based database applications for real world problems
CO5. Implement Relational algebra queries

)

Total Hours: 56 Hrs

Theory (14 Hours)
Study of Postgres SQL, PL/SQL programming and JDBC. Concepts of views, scripts, triggers and transactions, SQL DBA, PHP, Eclipse. Servlets

Practical (42 Hours)
Laboratory exercises which include defining schemas for applications, creation of a databases, writing SQL and PL/SQL queries, to retrieve information from the databases, use of host languages, interface with embedded SQL, use of forms & report writing packages available with the chosen RDBMS product preferably Postgres SQL Programming exercises on using scripting languages like PHP, Giving web interfaces for back end database applications.
Exercises on Programming in Java for connecting Postgres SQL databases using JDBC.
Exercises on creating web page interfaces for database applications using servlets.

References:

1. Avi Silberschatz, Hank Korth, and S. Sudarshan, Database System Concepts, (5/e), McGraw Hill, 2005
2. R. Elmasri and S. Navathe, Fundamentals of Database Systems, Addison Wesley , (5/e) , 2007

CS3096 COMPUTATIONAL INETELLIGENCE LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)
State Space Search, Two-agent Games, Logic, Machine Learning

Practical (42 Hours)
State Space Search – Water Jug Problem, Missionaries and cannibals, Tower of HANOI, Eight puzzle, Implementation of these problems using both uninformed and informed search. – BFS, DFS, Best First Search, A*

Two-agent Games – Tic-Tac-Toe using Min-Max search and Alpha-Beta pruning, *Constraint Satisfaction Problems* – N-Queens using Heuristic repair and constraint propagation

Logic-Unification, Resolution,Answer Extraction Using Resolution

Machine Learning – Decision Tree, Candidate Elimination, Clustering (K-means), Neural net learning (Perceptron), Genetic algorithms (2SAT), Expert Systems, Natural Language Processing

References:

1.George F Luger, *Artificial Intelligence- Structures and Strategies for Complex Problem Solving*, 4/e, 2002, Pearson Education.
2. E. Rich, K.Knight, *Artificial Intelligence*, 2/e, Tata McGraw Hill
3. S Russel, P Norvig, *Artificial Intelligence- A Modern Approach*, 2/e, Pearson Education, 2002
4. Winston. P. H, *LISP*, Addison Wesley
5. Ivan Bratko, *Prolog Programming for Artificial Intelligence*, 3/e, Addison Wesley, 2000

CS3097 WEB PROGRAMMING LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)

Review of basic technologies and concepts in Web Programming

Practical (42 Hours)

- Basic web client: Client programming, processing and parsing data when reading from a network socket - basics of the HTTP protocol.
- Basic web server: Client-server programming - Implement a protocol. 1.0 specification of HTTP - conditional get and cookies.
- Concurrent web server: Modifying web server for pool of threads - semaphores to synchronize access to shared memory.
- Performance evaluation: Workload generation, and performance evaluation. performance improvement gained by using threads - optimization.
- Peer-to-peer web browser: Peer-to-peer programming – building a distributed system. Peer to peer file sharing – synchronization similar to BitTorrent tracker. Quantifying scalability.
- Complete web application: Developing a database-driven complete web application following SDLC. Database backend (say MySQL) – application in PHP / Rails.

References:

1. Sam Ruby, Dave Thomas and David Heinemeier Hansson. Agile Web Development with Rails (3/e), Pragmatic Programmers, 2009.
2. Hugh E. Williams and David Lane. Web Database Applications with PHP and MySQL (2/e), O'Reilly & Associates, May 2004

CS4091 BIOCOMPUTING LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56Hrs

Module 1 (3 (T) + 10 (P) Hours)
Familiarization with Bioinformatics Resources: Understanding of biological databases [GenBank, EMBL, DDBJ, PDB, PIR, SwissProt], Retrieving and analyzing various types of data from these databases, Study of sequence alignment tools (both standalone and online versions) [DotPlot, Clustal, BLAST, FASTA], Study of PHYLIP.

Module 2 (3 (T) + 10 (P) Hours)
Introduction to Bio-programming languages: BioPerl, BioPython, BioJava.

Module 3 (3 (T) + 10 (P) Hours)
Study of Genomics and Proteomics Tools: Working with Genscan, Study of molecular visualization tools [Rasmol, Deep View], Study of Protein structure prediction tools [SCOP, MODELLER, I-TASSER]

Module 4 (5 (T) + 12 (P) Hours)
Implementation of algorithms in Bioinformatics: Sequence analysis and alignment, Motif finding, Protein structure prediction, Construction of Phylogenetic trees.

References:

1. Neil C Jones and Pavel A Pevzner, An Introduction To Bioinformatics Algorithms, MIT Press, August 2004.
2. Richard Ernest Bellman, Dynamic Programming, Princeton University Press, 2003.
3. Dan Gusfield, Algorithms On Strings, Trees, And Sequences, Cambridge University Press, 1997.
4. Gary Benson and Roderic Page, Algorithms In Bioinformatics, Springer, Vol 2812, 2003.

CS4092 DATA MINING LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours) + Practical (42 Hours)

Introduction to Scilab Matrix operations, Plotting functions, contours (2(T)+6(P)Hours)
Classification Bayesian classifier, Perceptron , Support Vector Machine(3(T)+12(P) Hours)
Clustering K-means and EM Clustering (3(T)+6(P) Hours) Association rule mining (2(T)+6(P) Hours)
Feature selection (2(T)+6(P) Hours) Introduction to Weka (2(T)+6(P) Hours)

References:

1. Pang-Ning Tan,Michael Steinbach and Vipin Kumar Introduction to Data Mining, Pearson Education 2006.
2. Han and Kamber, Data Mining: Concepts and Techniques, (2/e), Morgan Kaufmann

CS4093 IMAGE PROCESSING LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

.

Total Hours: 56 Hrs

Theory (14 Hours)
An introduction to digital images- sampling, quantization. Basic image processing, arithmetic processing. Image enhancement and point operation. Image enhancement and spatial operation. Color images and models models. Frequency domain operations.

Practical (42 Hours)
Lab1: An introduction to digital images- sampling, quantization, Image re-sampling, Image properties: bit-depth
Lab2: Basic image processing, arithmetic processing

Lab3: Image enhancement and point operation- Linear point operation, clipping, thresholding, negation, non-linear mapping, intensity slicing, image histogram, histogram equalization.

Lab4: Image enhancement and spatial operation- Convolution, correlation, linear filtering, edge detection.
Lab5: Color images- color models, color enhancement, color thresholding.
Lab6: Frequency domain operations- fourier transform, freq domain filtering

References:

1. Rafael C., Gonzalez & Woods R.E., Digital Image Processing, Addison Wesley, 2007.
2. Jain A.K, Fundamentals of Digital Image Processing, Prentice Hall, Englewood Cliffs, 2002.
3. Schalkoff R. J., Digital Image Processing and Computer Vision, John Wiley, 2004.

CS4094 COMPUTER VISION LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)

Edge operations: Various edge operators.
Segmentation and clustering techniques and applications.
Colouring and color image processing. Object detection and classification.
Computation of 3D scene from 2D.

Practical (42 Hours)
MatLab implementation for the following:
1. Edge operations:
2. Segmentation: by clustering, segmentation by fitting models-Vision applications.
3. Colouring techniq ues, Pseudo-colouring,
4. Colour image analysis.
5. Object detection and classifications
6. Computation of 3D scene from 2D.

References:

1. David A Forsynth and Jean Ponce (2003), Computer Vision- A modern approach, Pearson education series, 2003.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle (2008), Digital image processing and computer vision, Cengage learning, 2008
3. Schalkoff R. J., Digital Image Processing and Computer Vision, John Wiley, 2004.

CS4095 COMPUTER GRAPHICS LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)
OpenGL programming - constructs and standards.

Practical (42 Hours)
Drawing Geometric Primitives - case studies.
Create simple models.
Interactive Transformations and Projections
Parsing simple mesh file formats
Rendering meshes.
Case Study: Model a scene, Place lights on the scene, render shadows and texture models.

References:

1. D. Shreiner, M. Woo, J. Neider and T. Davis, OpenGL Programming Guide, Addison Wesley, 2005.

CS4096 SOFTWARE ENGINEERING LABORATORY

Pre-requisite: CS3004 Software Engineering

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)
Introductory Lectures on the use of appropriate tools is to be given.
Peer review discussions of deliverables will also be done in theory sessions.

Practical (42 Hours)
Objective is to develop a significant software product using sound software engineering principles by small student groups. Choice of appropriate methodology and standard tools are also expected. The lab will have deliverables at each milestone of development.

1. Problem Statement / Product Specification
2. Project Plan – Project Management Tool to be identified and Estimation and Costing to be done.
3. Requirements Document – Specification Tool choice to be justified - In class Review
4. Design Document – Choice of Methodology to be justified - In class Review
5. Code and Test Report – Peer review documents of standards adherence to be provided
6. Demo – Integrated Product or Solution to the problem
7. Review of the process and analysis of variation from initial plan and estimation.

References:

1. Roger S Pressman, Software Engineering: A Practitioner's Approach (6/e.), Mc Graw Hill, 2008.

CS4097 OBJECT ORIENTED PROGRAMMING LABORATORY

Pre-requisite: Nil

ELECTIVE

L	T	P	C
1	0	3	3

Total Hours: 56 Hrs

Theory (14 Hours)
Procedural vs. Objected oriented approaches – Concept of Abstraction - Design and analysis using OO methodologies. Introduction to UML.

Practical (42 Hours)
The implementation has to be done using languages like C++/Java/C#.
Programs to study

Functions – Control structures – String handling – File handling
Error and Exception handling
Class – Objects –Instantiation
Principles of Inheritance, Encapsulation, Polymorphism – Overloading, Virtual functions
OO Design with stress on interface specification. Automated code generation and component reuse - UML

References:

1. B Stroustrup, The C++ Programming Language (3/e). Addison Wesley, 1997.
2. Steve Oualline, Practical C++ Programming (2/e). O'Reilly & Associates, 2002.
3. J Nino and F A Hosch, An introduction to programming and object oriented design using Java. Wiley India, 2010

Course Assessment Methods

The method of assessment for any course is decided by the concerned faculty in consultation with the class committee and announced to the students in the beginning of the semester. We adopt a continuous evaluation policy for all lecture and practical courses. The evaluation consists of two tests and an end semester examination. A minimum of 40% weightage is assigned to the end semester examination. Continuous evaluation may include assignments or quizzes. The results are discussed in class committee consisting of a committee chairman, faculty and students.

The assessment methods and weightage for each course is tabulated below.

Code	Course	Test I	Test II	End Semester Exam	Assign ments	Lab Exerc ises	Qui z	Projec ts	Presentations	Total
CS 1001	FOUNDATIONS OF COMPUTING	15	15	50	20	-	-	-	-	100
ZZ1004	COMPUTER PROGRAMMING	15	15	50	-	20	-	-	-	100
MA2001	MATHEMATICS III	25	25	50	-	-	-	-	-	100
CS2001	LOGIC DESIGN	25	25	50	-	-	-	-	-	100
CS2002	FOUNDATIONS OF PROGRAMMING	25	25	50	-	-	-	-	-	100
EC2014	SIGNALS AND SYSTEMS	20	20	50	10	-	-	-	-	100
CS2091	LOGIC DESIGN LABORATORY	30(Viva) + 40			-	30		-	-	100
CS2092	PROGRAMMING LABORATORY	25 * 4			-	-	-	-	-	100
MA2002	MATHEMATICS IV	20	20	50	10	-	-	-	-	100
CS2004	COMPUTER ORGANIZATION	20	20	50	10	-	-	-	-	100
CS2005	DATA STRUCTURES AND ALGORITHMS	20	20	40	20	-	10	-	-	100
CS2006	DISCRETE STRUCTURES	30	30	40	-	-	-	-	Moodle10%	Round ed to 100%
CS2093	HARDWARE LABORATORY	20	20	30	-	30	-	-	-	100
CS2094	DATA STRUCTURES LABORATORY	20	20	20	25		15	-	-	100
CS3001	THEORY OF COMPUTATION	30	30	40	-	-	-	-	-	100

CS3002	DATABASE MANAGEMENT SYSTEMS	20	20	50	10	-	-	-	-	100
CS3003	OPERATING SYSTEM	20	20	50	10	-	-	-	-	100
CS3004	SOFTWARE ENGINEERING	15	15	50	-	-	10	10	-	100
CS3005	COMPILER DESIGN	25	25	50	-	-	-	-	-	100
CS3006	COMPUTER NETWORKS	20	20	50	10	-	-	-	-	100
CS4001	ENVIRONMENTAL STUDIES	-	-	-	-	-	-	-	25*4	100
MS4003	ECONOMICS	20	20	50	-	-	10	-	-	100
ME4104	PRINCIPLES OF MANAGEMENT	20	20	50	10	-	-	-	-	100
CS4021	NUMBER THEORY AND CRYPTOGRAPHY	15	15	50	15	-	5	-	-	100
CS4022	PRINCIPLES OF PROGRAMMING LANGUAGES	15	15	50	10	-	10	-	-	100
CS4023	COMPUTATIONAL INETELLIGENCE	15	15	50	20	-	-	-	-	100
CS4024	INFORMATION THEORY	30	30	40	-	-	-	-	-	100
CS4025	GRAPH THEORY AND COMBINATORICS	25	25	40	10	-	-	-	-	100
CS4026	COMBINATORIAL ALGORITHMS	30	30	40	-	-	-	-	-	100
CS4027	TOPICS IN ALGORITHMS	30	30	40	-	-	-	-	-	100
CS4028	QUANTUM COMPUTATION	30	30	40	-	-	-	-	-	100
CS4029	TOPICS IN THEORY OF COMPUTATION	30	30	40	-	-	-	-	-	100
CS4030	COMPUTATIONAL COMPLEXITY	30	30	40	-	-	-	-	-	100
CS4031	COMPUTATIONAL ALGEBRA	30	30	40	-	-	-	-	-	100
CS4032	COMPUTER ARCHITECTURE	20	20	50	10	-	-	-	-	100
CS4033	DISTRIBUTED COMPUTING	20 + 50			10	-	5	10	5	100
CS4034	MIDDLEWARE TECHNOLOGIES	-	-	70	10	-	5	10	5	100
CS4035	COMPUTER SECURITY	15	15	50	5	-	-	10	5	100
CS4036	ADVANCED DATABASE MANAGEMENT	20	20	50	10	-	-	-	-	100

	SYSTEMS									
CS4037	CLOUD COMPUTING	20	20	50	10	-	-	-	-	100
CS4038	DATA MINING	20	20	50	10	-	-	-	-	100
CS4039	MULTI AGENT SYSTEMS	15	15	50	15	-	5			100
CS4040	BIOINFORMATICS	20	20	50	10	-	-	-	-	100
CS4041	NATURAL LANGUAGE PROCESSING	15	15	50	10	-	10	-	-	100
CS4042	WEB PROGRAMMING	15	15	50	-	-	-	20	-	100
CS4043	IMAGE PROCESSING	15	15	50	-	20	-	-	-	100
CS4044	PATTERN RECOGNITION	15	15	50	-	20	-	-	-	100
CS4045	MEDICAL IMAGE PROCESSING	15	15	50	-	20	-	-	-	100
CS4046	COMPUTER VISION	15	15	50	-	20	-	-	-	100
CS4047	COMPUTER GRAPHICS	20	20	50	10	-	-	-	-	100
CS4048	TOPICS IN COMPILERS	25	25	50	-	-	-	-	-	100
CS4049	ADVANCED COMPUTER NETWORKS	15	15	50	10	-	10	-	-	100
CS4050	DESIGN AND ANALYSIS OF ALGORITHMS	20	20	40	10	-	10	-	-	100
CS4051	CODING THEORY	30	30	40	-	-	-	-	-	100
CS4052	LOGIC FOR COMPUTER SCIENCE	25	25	40	10	-	-	-	-	100
CS3091	COMPILER LABORATORY	-	-	20	-	70	10	-	-	100
CS3092	OPERATING SYSTEMS LABORATORY	-	-	20	-	70	10	-	-	100
CS3093	NETWORKS LABORATORY	-	-	-	-	90	10	-	-	100
CS3094	PROGRAMMING LANGUAGES LABORATORY	-	-	-	-	20 * 5	-	-	-	100
CS3095	DATABASE MANAGEMENT SYSTEMS LABORATORY	-	-	-	-	90	10	-	-	100
CS3096	COMPUTATIONAL INETELLIGENCE LABORATORY	20 * 5			-	-	-	-	-	100

CS3097	WEB PROGRAMMING LABORATORY	20 * 5			-	-	-	-	-	100
CS4091	BIOCOMPUTING LABORATORY	-	-	-	-	90	10	-	-	100
CS4092	DATA MINING LABORATORY	-	-	-	-	90	10	-	-	100
CS4093	IMAGE PROCESSING LABORATORY	30			-	70	-	-	-	100
CS4094	COMPUTER VISION LABORATORY	30			-	70	-	-	-	100
CS4095	COMPUTER GRAPHICS LABORATORY	30			-	70	-	-	-	100
CS4096	SOFTWARE ENGINEERING LABORATORY	20 * 5			-	-	-	-	-	100
CS4097	OBJECT ORIENTED PROGRAMMING LABORATORY	25 * 4			-	-	-	-	-	100
CS4098	SEMINAR	15 (viva)	-	-	-	-	-	-	30 + 30 (Report) + 10 (Topic) + 15(Slides)	100
CS4089	PROJECT								20(Report) + 80(Presentations)	100
CS4099	PROJECT								20(Report) + 80(Presentations)	100